→ Illumination models:

1) Light sources → Light emitting sources (sun, Light bulb)
2) Light reflected sources → such as walls

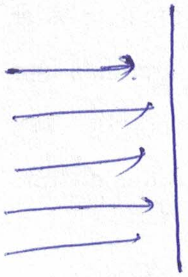
⇒ 1) Point source
2) Diffuse Reflection
3) Specular Reflection → depends on Surface optical Parameter

such as → glossy
- matte
- glass
- opaque.

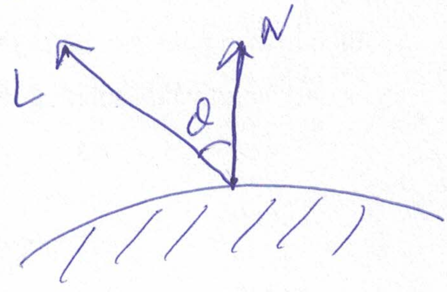
1) Ambient Light. → A surface that is not exposed directly to a light source still will be visible if nearly objects are illuminated.

→ surface is then illuminated with constant value (I_a). The resulting reflected Light is constant for each surface.

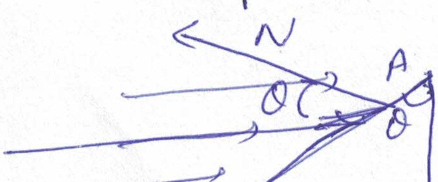
$$I_{amb} = K_a I_a$$



$$I_{diff} = K_d I_l \cos\theta = K_d I_l (N \cdot L)$$



To Light source.

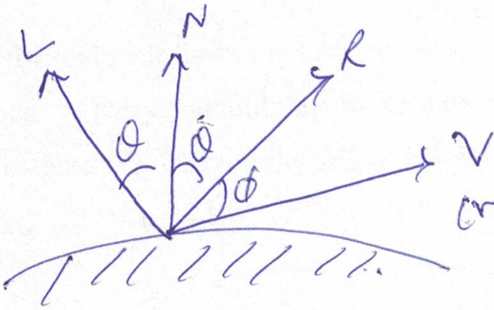


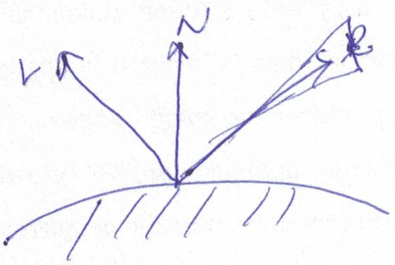
Also

3) Specular Reflection & Phong Model:

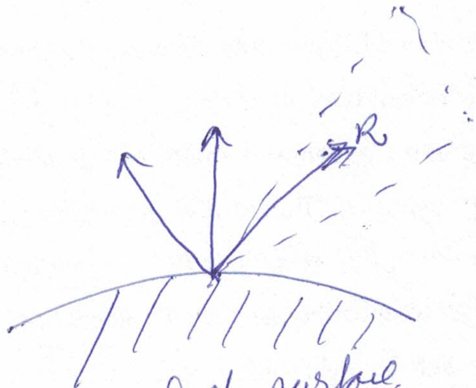
$I_{spec} = W(\theta)\, I_l \cos^{n_s}\phi$

$n_s \rightarrow$ specular parameter





(Shiny surface large n_s)



Dull surface (small n_s)

Halftone Patterns & Dithering Techniques:

⇒ Halftone Patterns

→ When an o/p device has limited intensity range.

Ex → Bilevel Monitors & Printers

↑ Produce pictures that appear to be displayed with multiple intensity values.

⇒ Continuous-tone photographs are reproduced for publication in newspapers, magazines & books with a printing process is called halftoning, & reproduced pictures are

⇒ In Computer Graphics, halftone reproductions are approximated using rectangular pixel regions, called halftone patterns or pixel patterns.

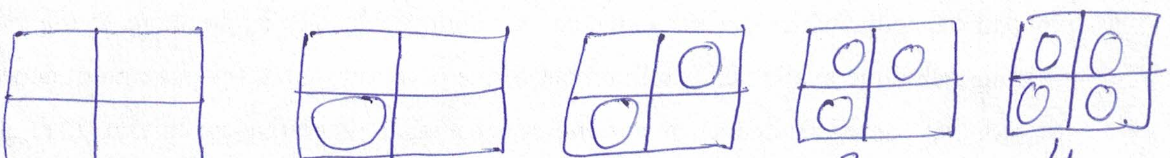
→ with n by n pixels for each grid on a bilevel system, (n^2+1) intensity levels can be represented.

Ex: if $n=2$ 2 by 2

$n^2+1=5$ ⇒ no. of intensity levels.

Off → pattern 0

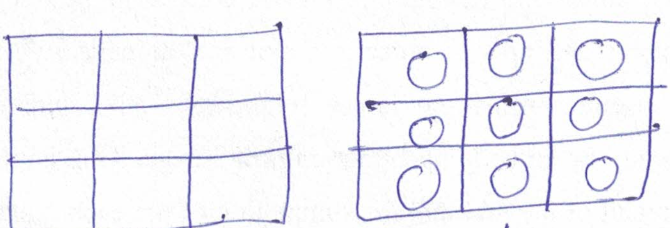
On → pattern 1



0   1   2   3   4

$0 \le I \le 0.2$ $0.2 \le I < 0.4$ $0.4 \le I < 0.6$ $0.6 \le I < 0.8$ $0.8 \le I \le 1$

⇒ if $n=3$, $n^2+1=10$ ⇒ intensity levels.



0   1

$0 \le I \le 0.1$ $0.1 \le I < 0.2$ $0.2 \le I < 3$ $0.2 \le I < 0.4$

⇒ Dithering Techniques:

⇒ Refers to techniques for approximating halftones without reducing resolution.

⇒ Dither Noise → Random value, added to pixel intensities

1) Ordered Dither Method:

a) Generates intensity variations with a one-to-one mapping of points in a scene to the display pixels.

b) To obtain n^2 intensity levels, set n by n dither matrix D_n, whose elements are distinct positive integers in the range of $n^2 - 1$.

For Ex. $D_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$

& $D_3 = \begin{bmatrix} 7 & 2 & 6 \\ 4 & 0 & 1 \\ 3 & 8 & 5 \end{bmatrix}$

elements of D_2 & D_3 are in same order as the pixel mask for setting up 2 by 2 & 3 by 3 pixel grids.

c) Determine display intensity values by comparing input intensities to the matrix elements. Each input intensity is first scaled to the range $0 < I \leq n^2$.

d) If the intensity I is to be applied to screen position (x, y), calculate row & column numbers for the dither

⇒ If $I > D_n(i,j)$, turn on pixel at position (x,y), otherwise, turn off.

⇒ The no. of intensity levels is taken to be multiple of 2, higher order dither matrix are then obtained from lower order matrices with recurrence relation

$$D_n = \begin{bmatrix} 4D_{n/2} + D_2(1,1)U_{n/2} & 4D_{n/2} + D_2(1,2)U_{n/2} \\ 4D_{n/2} + D_2(2,1)U_{n/2} & 4D_{n/2} + D_2(2,2)U_{n/2} \end{bmatrix}$$

assuming $n \geq 4$. $U_{n/2} \rightarrow$ Unity matrix

$$D_4 = \begin{bmatrix} 4D_2 + D_2(1,1)U_2 & 4D_2 + D_2(1,2)U_2 \\ 4D_2 + D_2(2,1)U_2 & 4D_2 + D_2(2,2)U_2 \end{bmatrix}$$

$$\begin{bmatrix} 12 & 4 \\ 0 & 8 \end{bmatrix} + \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} \quad \begin{bmatrix} 12 & 4 \\ 0 & 8 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

~~16~~ ~~8~~

$$D_4 = \begin{bmatrix} 15 & 7 & 13 & 5 \\ 3 & 11 & 1 & 9 \\ 12 & 4 & 14 & 6 \\ 0 & 8 & 2 & 10 \end{bmatrix}$$

2) <u>Error Diffusion</u>: Another method for mapping a picture with m by n points to a display area with m by n pixels is error diffusion.

⇒ Here the error b/w an input intensity value and the displayed pixel intensity level at a given position is dispersed, or diffused, to pixel positions to the right & below the current pixel position.

1) Scan across the rows of M from left to right, top to bottom, & determine the nearest available pixel intensity level for each element of M.

2) Then the error b/w the values stored in matrix M & the displayed intensity level at each pixel position is distributed to neighbouring elements in M, using following

<u>Algo</u>:

```
for (i=0 ; i<m ; i++)
  for (j=0; j<n ; j++)
{
    Ii,j := Ik   → available intensity level
    err := Mi,j - Ii,j
    Mi,j+1 := Mi,j+1 + α.err
    Mi+1,j-1 := Mi+1,j-1 + β.error
                 Mi+1,j + γ.error
```

⇒ where $\alpha + \beta + \gamma + \delta \leq 1$

Error diffusion parameters → $(7/16, 3/16, 5/16, 1/16)$ → are choice to select values of $\alpha, \beta, \gamma, \& \delta$.

⇒ Another variation of error diffusion is <u>dot diffusion</u>.