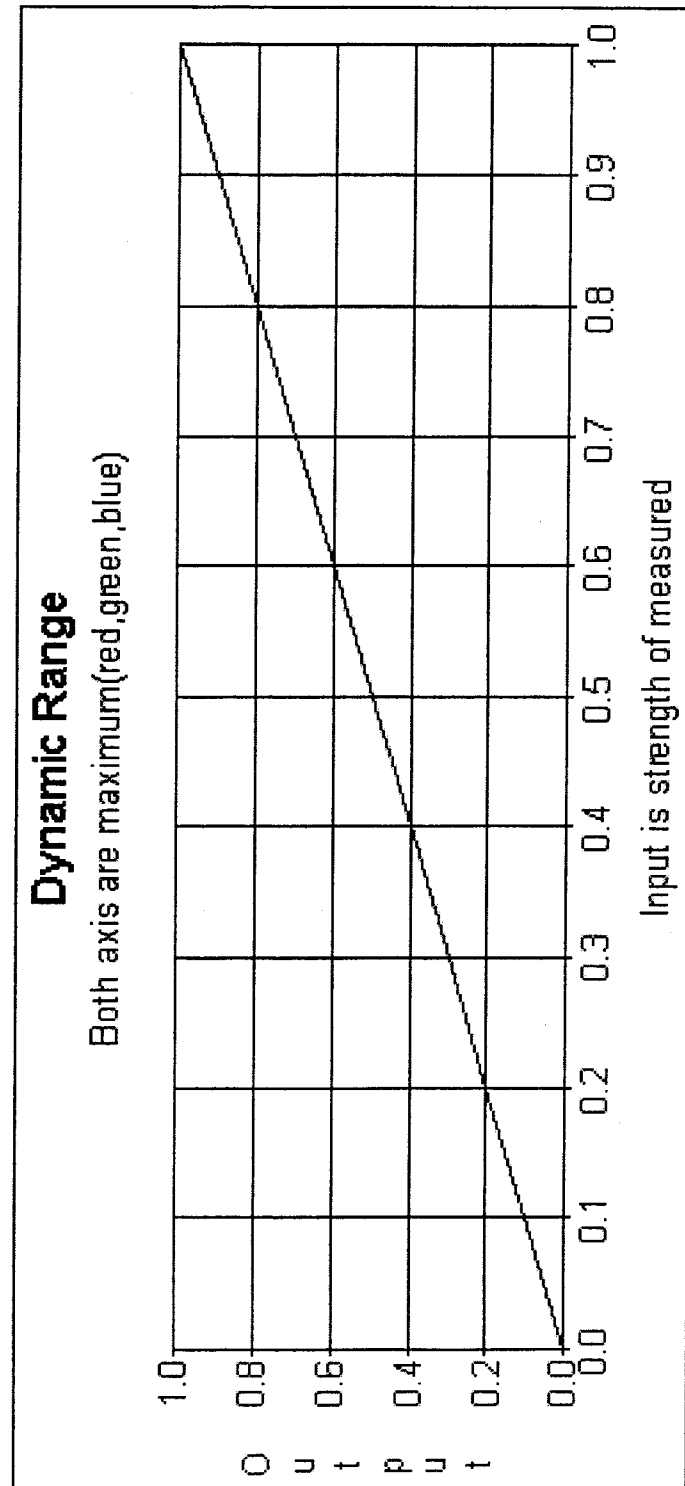


1/15

**Fig. 1**

## 2/15

**pbox.Picture = LoadPicture(file\_in)**

a rectangular picture of columns and rows is displayed

**For icol = 1 To pbox.ScaleWidth**

loops through the columns of the image,

**For irow = 1 to pbox.ScaleHeight**

loops through the rows of the image

**color = pbox.Point(icol,irow)**

the color of the point at current column, row.

**blue = color / 65536**

This sets blue to the most significant byte (division by  $256 * 256$ )

**color = color mod 65536**

This keeps only the 2 least significant bytes (mod of  $256 * 256$ )

**green = color / 256**

This sets green to the (next) most significant byte (division by 256)

**red = color mod 256**

This sets red to the least significant byte (modulus of 256)

**strength = red**

**If strength < green Then strength = green**

**If strength < blue Then strength = blue**

the dot maximum is found

**corr = corra(strength)**

Lookup correction for that value of dot maximum then correct RGB in the dot

**red = red \* corr**

**green = green \* corr**

**blue = blue \* corr**

Red, green, blue are now corrected to desired values

**pbox.PSet(icol,irow),RGB(red,green,blue)**

Put the adjusted and corrected dot back into the image on the screen

**Next irow**

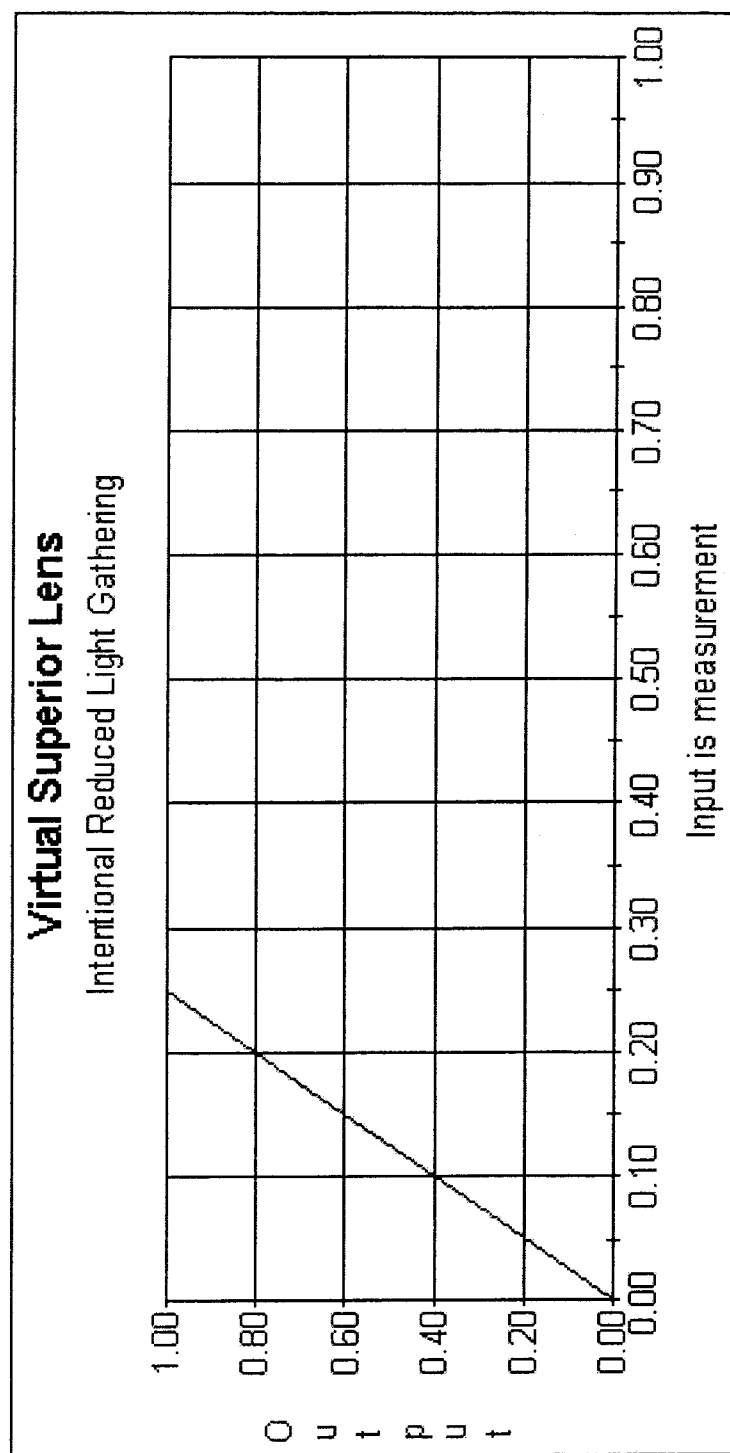
Get next row dot in the column

**Next icol**

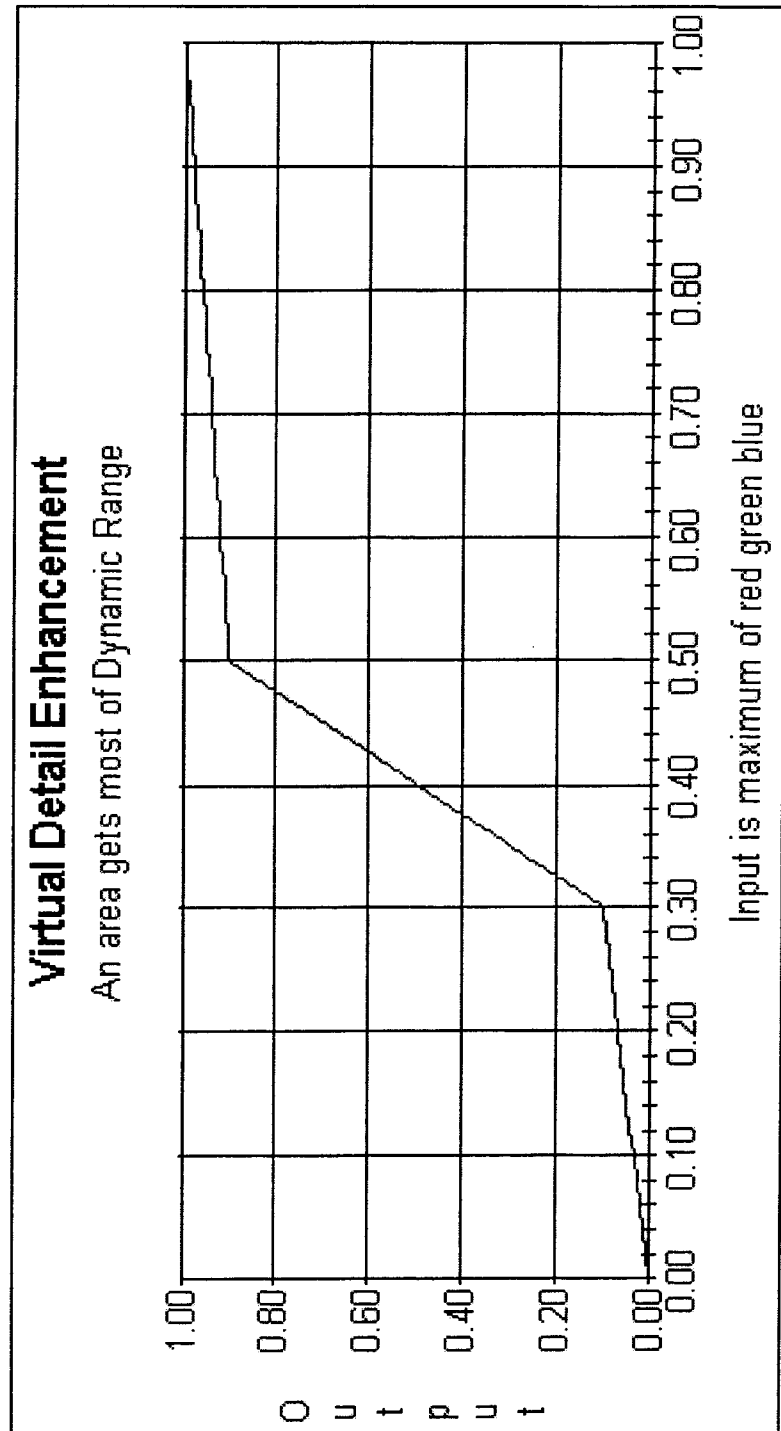
Get next image column of dots

**Fig. 2**

3/15

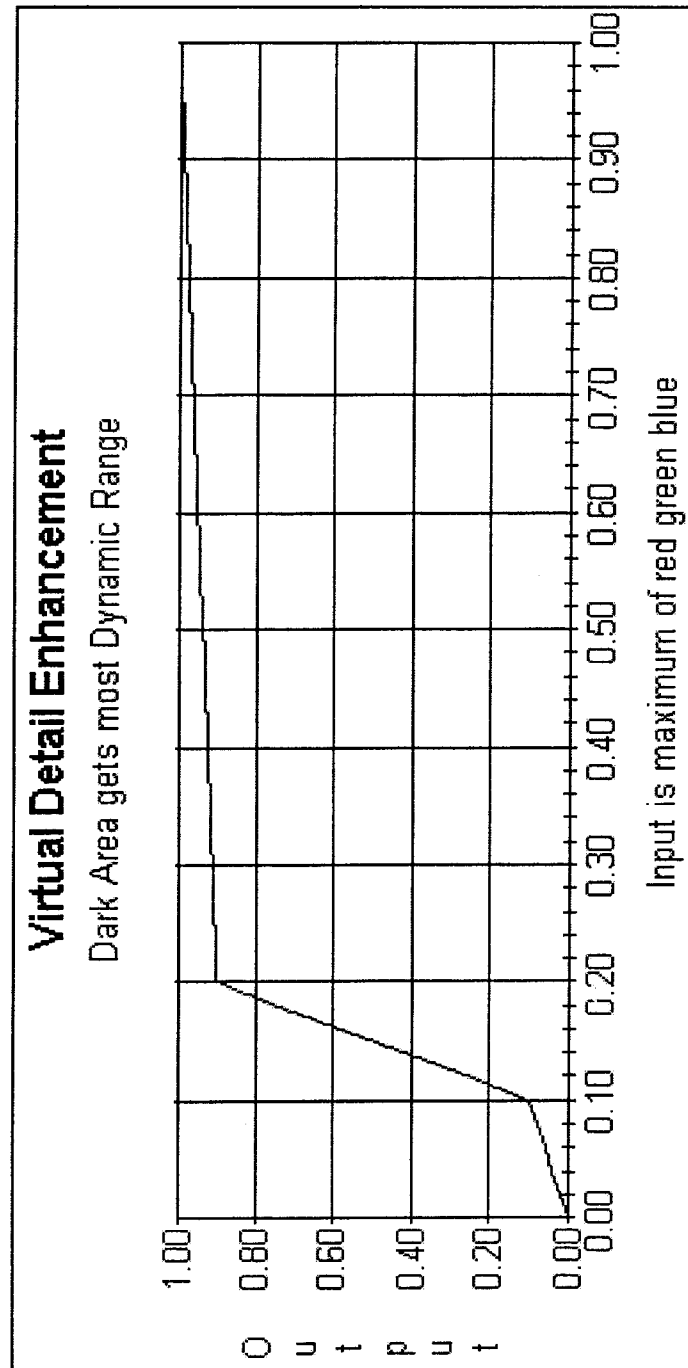
**Fig. 3**

4/15

**Fig. 4**

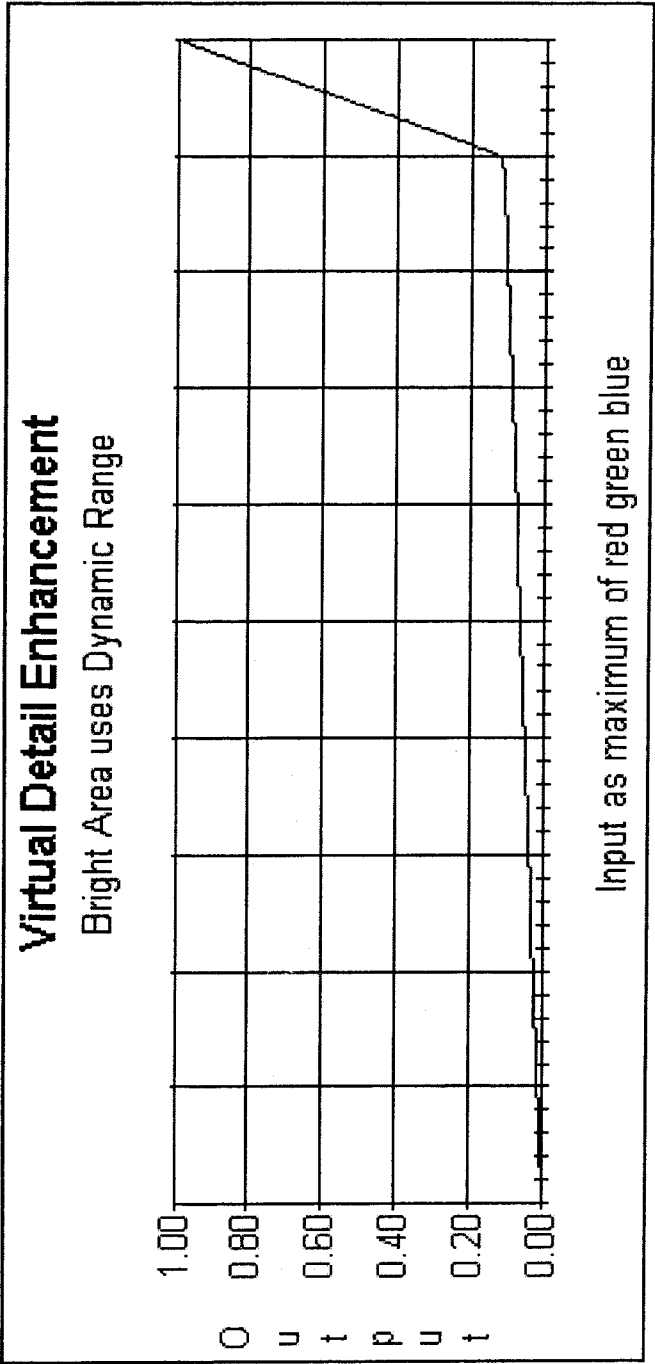
5/15

Fig. 5



6/15

Fig. 6



## 7/15

Assuming the area was selected so that xdown is less than xup

**x1 = xdown +1**

**x2 = xup -1**

**y1 = ydown +1**

**y2 = yup -1**

This Statement Ends the loop limit determination

**For icol = x1 To x2**

**For irow = y1 to y2**

The next 5 lines determine the color values (RGB)

**color = pbox.Point(icol,irow)**

**blue = color / 65536**

**color = color mod 65536**

**green = color / 256**

**red = color mod 256**

Find the dot Maximum

**strength = red**

**If strength < green Then strength = green**

**If strength < blue Then strength = blue**

The next line builds the histogram

**hist(strength) = hist(strength) + 1**

hist is an array (look up table) that stores the histogram counts

**Next irow**

**Next icol**

Find the minimum and maximum strengths, in some statistical sense, recorded stored in the histogram look up table (the array hist()). Creating a second look up table with the running totals of the histogram:

**hsum(0) = hist(0)**

hsum is the look up table used to hold the running total of the histogram

first (zeroth) running total is the first (zeroth) histogram element

**For indx = 1 To drmx**

indx is an index that runs from 1 (not 0) to the dynamic range maximum

**hsum(indx) = hsum(indx - 1) + hist(indx)**

the running total is calculated as previous running total plus next value

**Next indx**

**Fig. 7a**

## 8/15

Code for histogram look-up table

**hsum(0) = hist(0)**

hsum is the look up table used to hold the running total of the histogram

first (zeroth) running total is the first (zeroth) histogram element

**For indx = 1 To drmx**

Indx is an index that runs from 1 (not 0) to the dynamic range maximum

**hsum(indx) = hsum(indx - 1) + hist(indx)**

the running total is calculated as previous running total plus next value

**Next indx**

**p002 = 0.02 \* hsum(drmx) wherein**

result is 2% of the total count and hsum(drmx) is the last running total (the total) of the histogram counts

**p098 = 0.98 \* hsum(drmx)**

result is 98% of the total count, The next step is to find the index values corresponding to the 2% and 98% occurrence in the histogram.

**i002 = -1; i098 = -1**

these variables will hold the indices at the 2% and 98% level, for now they are set to -1

**For indx = 0 to drmx**

the loop runs over all the strength values

**if hsum(indx) < p002 then i002 = indx**

when the running total is less than 2% total then store it (stop storing it when it is over 2%)

**if hsum(indx) < p098 then i098 = indx**

when the running total is less than 98% total then store it (stop storing it when it is over 98%)

**Next indx**

Now, the histogram minimum and maximum values are set with a check to ensure that they are reasonable.

**hmin = i002 + 1**

hmin is set to 2% level strength index + 1 for reasons of symmetry

**If hmin < 1 then hmin = 1**

hmin is forced to be no less than 1

**hmax = i098**

hmax is set to 98% occurrence level of strength index

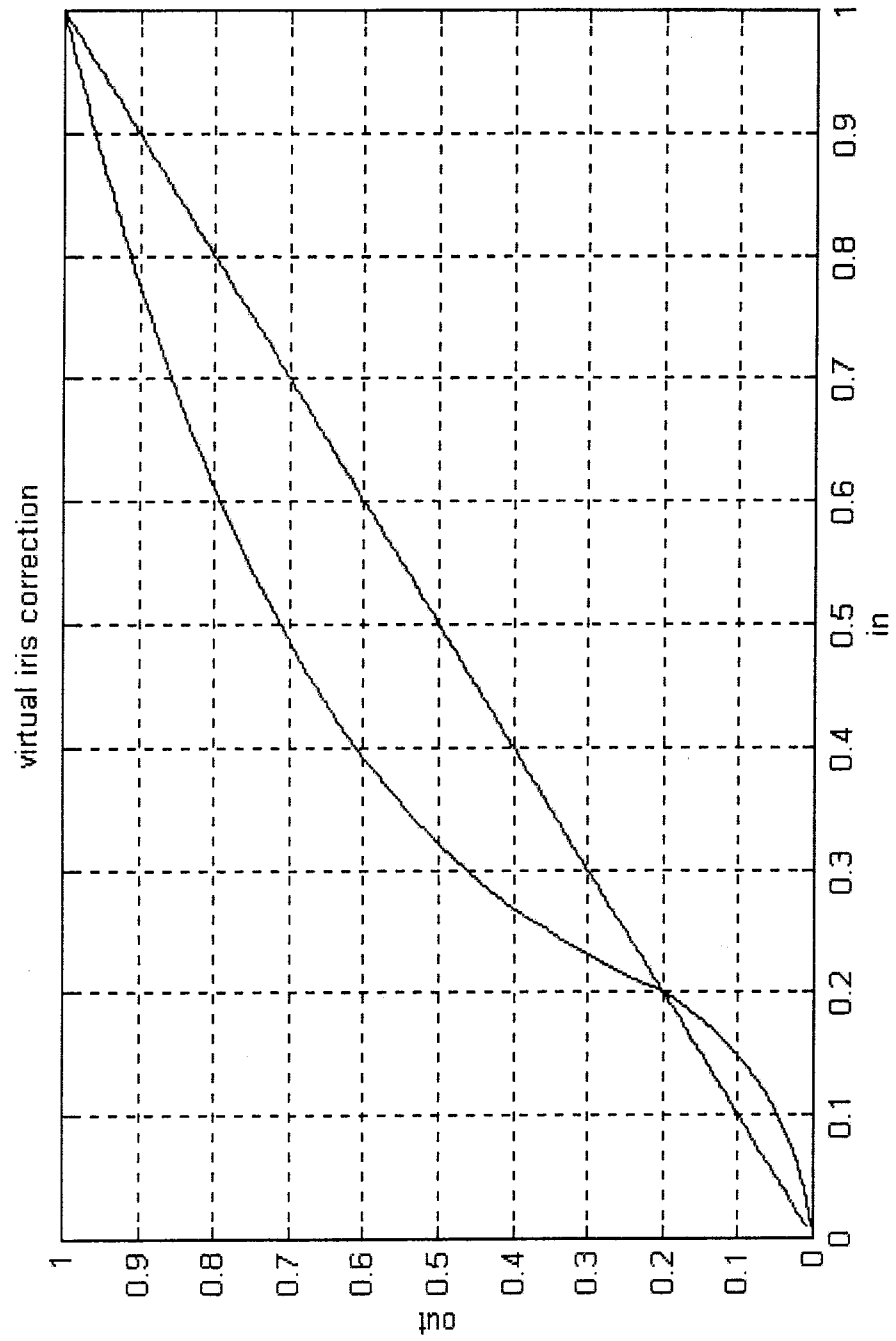
**If hmax > drmx - 1 then hmax = drmx - 1**

hmax is forced to be no more than dynamic range limit - 1 (254)

**Fig. 7b**

9/15

**Fig. 8**



**10/15**

**Fig. 9a**

Input Image

**Fig. 9b**

PRIOR ART

Brightened

**Fig. 9c**

PRIOR ART

Bright 80%  
Contrast

**Fig. 9d**

PRIOR ART

Bright 80%  
Saturation

**Fig. 9e**

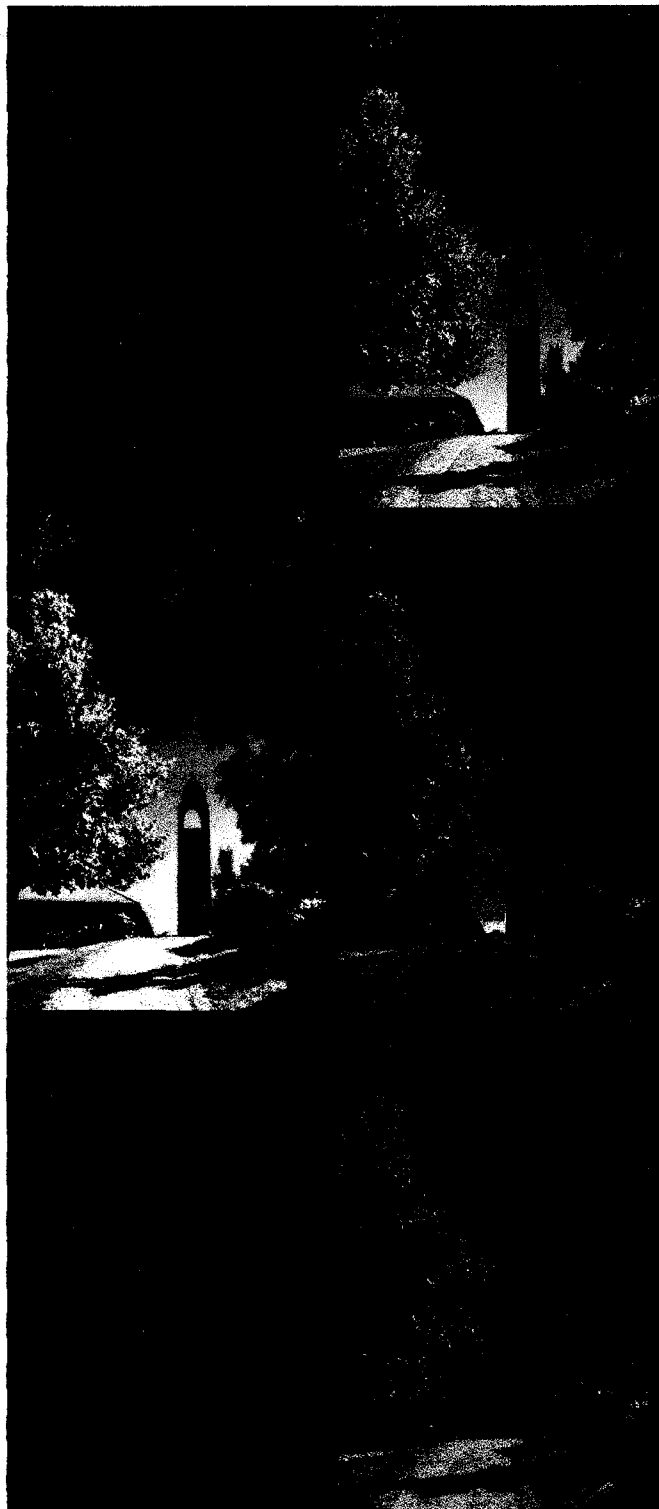
Virtual Flash

Output Image  
Full Dyn Range

**Fig. 9f**

Virtual Flash

Output Image  
Apply Fig. 8  
curve



11/15

**Fig. 10a**

Input Image



**Fig. 10b**

PRIOR ART  
Brightened



**Fig. 10c**

PRIOR ART  
Bright 60%  
Contrast



**Fig. 10d**

PRIOR ART  
Bright 60%  
Saturation 15%



**Fig. 10e**

Virtual

Output Image  
Full Dyn  
Range



**Fig. 10f**

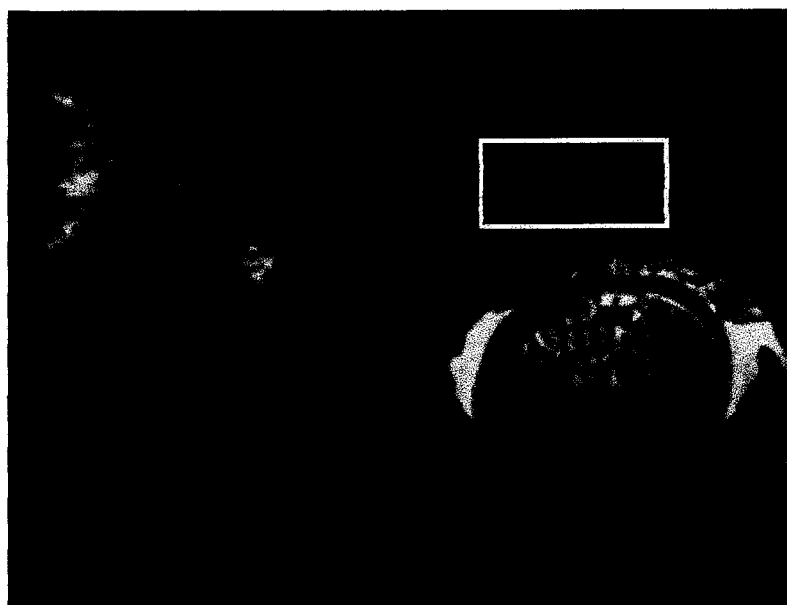
Virtual Flash  
Output Image  
Apply Fig. 8  
curve



**12/15**

**Fig. 11a**

Input Image



**Fig. 11b**

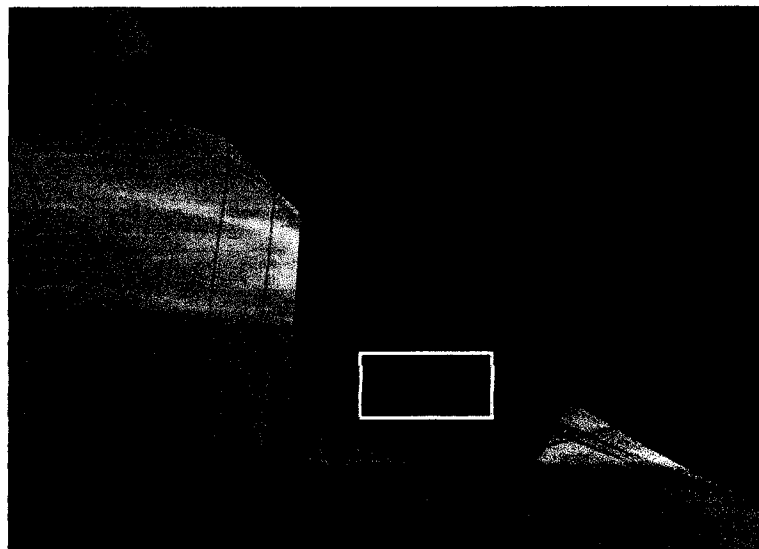
Forensic Flash  
Output Image



**13/15**

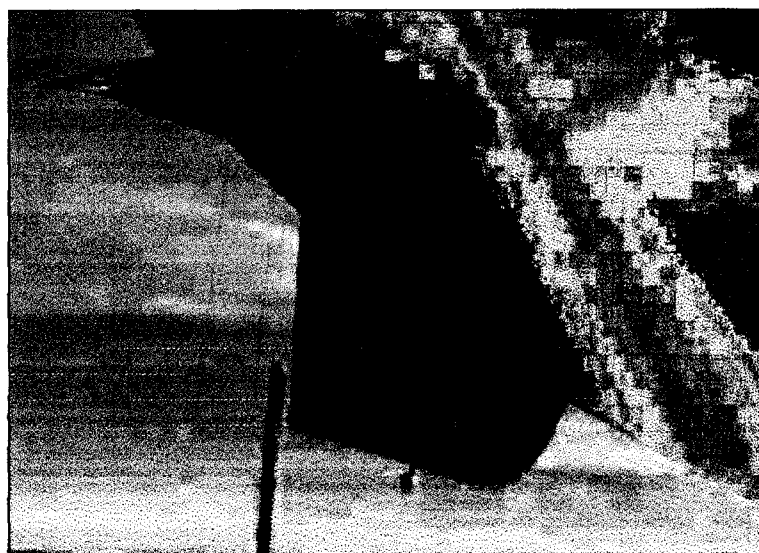
***Fig. 12a***

Input Image

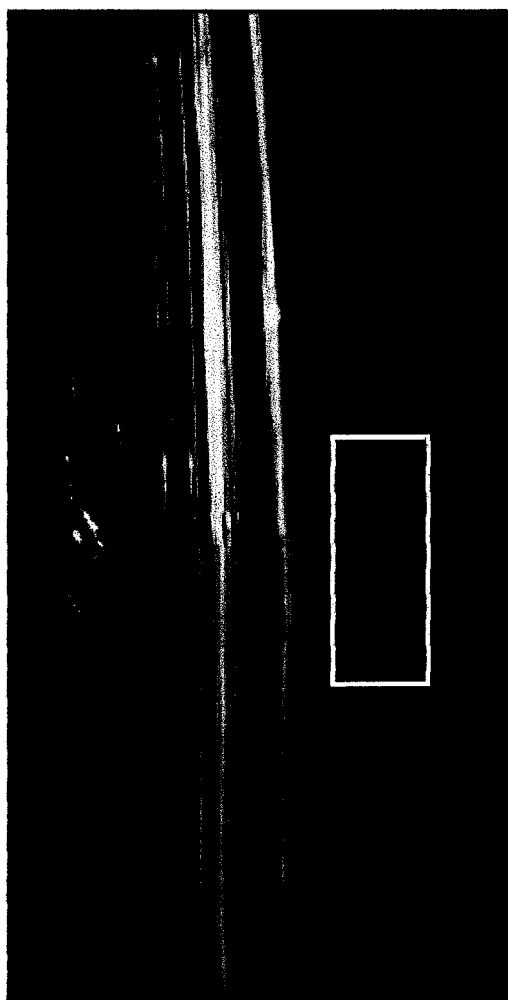


***Fig. 12b***

Forensic Flash  
Output Image



14/15



**Fig. 13a**

Input Image



**Fig. 13b**

Forensic Flash

Output Image

**15/15**

**Fig. 14a**

Input Image



**Fig. 14b**

Forensic Flash

Output Image  
Pick Snow



**Fig. 14c**

Forensic Flash

Output Image  
Pick Face

