|  |  |  |
| --- | --- | --- |
| Add A=B+k ; k=0,1,2,…,254 | Same as Increment++ | | |
| Frame=000 | Frame=128 | Frame=254 |
|  |  |  |
| Constant addition is a linear image operation that increases the constrast of the image, by shift pixels from its initial RGB by k towards pure white. Used to adjust constrast of image. | | |

|  |  |  |
| --- | --- | --- |
| Sub A=B-k ; k=0,1,2,…,254 | Same as decrement-- | | |
| Frame=000 | Frame=128 | Frame=254 |
|  | C:\Users\tenet\Desktop\Results\SUB\SUB 128 1.jpg | C:\Users\tenet\Desktop\Results\SUB\SUB 254 1.jpg |
| Constant subtraction is a linear image operation that decreases the constrast of the image, by shift pixels from its initial RGB by k towards pure black. Used to adjust constrast of image. | | |

|  |  |  |
| --- | --- | --- |
| Mul A=kB; k=0,1,2,…,254 | | |
| Frame=001 | Frame=005 | Frame=010 |
| C:\Users\tenet\Desktop\Results\MUL\MUL 1 1.jpg | C:\Users\tenet\Desktop\Results\MUL\MUL 5 1.jpg | C:\Users\tenet\Desktop\Results\MUL\MUL 10 1.jpg |
| Constant multiplication is a linear image operation that increases the brightness of the image, by scaling pixels its initial RGB by k. Used to adjust brightness of image. DARK 🡪 BRIGHT | | |

|  |  |  |
| --- | --- | --- |
| Div A = B/k ; k=1,2,…,254 | | |
| Frame=001 | Frame=005 | Frame=010 |
| C:\Users\tenet\Desktop\Results\MUL\MUL 1 1.jpg | C:\Users\tenet\Desktop\Results\DIV\DIV 5 1.jpg | C:\Users\tenet\Desktop\Results\DIV\DIV 10 1.jpg |
| Constant division is a linear image operation that decreases the brightness of the image, by scaling pixels its initial RGB by k. Used to adjust brightness of image. It continues AND operation. BRIGHT 🡪 DARK | | |

|  |  |  |
| --- | --- | --- |
| Absolute Diff A=|C-B| with threshold=k ; k=0,1,2,…,254 | | |
| Frame=000 | Frame=128 | Frame=254 |
| C:\Users\tenet\Desktop\Results\ADIFF\ADIFF 40 1.jpg | C:\Users\tenet\Desktop\Results\ADIFF\ADIFF 128 1.jpg | C:\Users\tenet\Desktop\Results\ADIFF\ADIFF 254 1.jpg |
| Absolute difference is a non linear linear image operation that subtract target image and background image, discards any pixel with length greator than threshold. This operation can be used for object detection by compering background frame and current frame. | | |

|  |  |  |
| --- | --- | --- |
| Diff A=C-B with threshold=k ; k=0,1,2,…,254 | | |
| Frame=000 | Frame=128 | Frame=254 |
| C:\Users\tenet\Desktop\Results\DIFF\DIFF 0 1.jpg | C:\Users\tenet\Desktop\Results\DIFF\DIFF 128 1.jpg | C:\Users\tenet\Desktop\Results\DIFF\DIFF 254 1.jpg |
| Difference operation is semilat to absolute difference operation, but does not absolute the resulting pixil. It is a non linear image operation that discards any pixel which length greator than threshold. It is also useful for object detection. | | |

|  |  |  |
| --- | --- | --- |
| Inverse A=~B= 255 - A; | | |
| Before |  | After |
| C:\Users\tenet\Desktop\Results\MUL\MUL 1 1.jpg |  | C:\Users\tenet\Desktop\Results\INV\INV 0 1.jpg |
| Inverse is a linear image operation that takes white pixel subtact it with current image pixel (i.e. RGB=255-RGB) . Same effect can be achieved with bitwise XOR of 255-1. It is useful to invert the image. | | |

|  |  |  |
| --- | --- | --- |
| Len A=B.length(); | | |
| Before |  | After |
| C:\Users\tenet\Desktop\Results\MUL\MUL 1 1.jpg |  | C:\Users\tenet\SkyDrive\VisualStudioSpace2015\OpenCVWorkspace\Tutorial1\Tutorial1\Results\LEN\LEN 1.jpg |
| Lengthining is a non linear image lossy operation that takes compute length of pixel, and override its values with length (i.e. R=G=B=Length), where Length = sqrt(R\*R + G\*G +B\*B). It is useful to grey the image. | | |

|  |  |  |
| --- | --- | --- |
| Rotation | | |
| ros, col |  | row, COLS-col |
| C:\Users\tenet\Desktop\Results\MUL\MUL 1 1.jpg  **Horizontal Reflection**  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ROT(rows-i,j) 1.jpg | **Vertical Reflection**  **Vertical Reflection** | **Diagonal Reflection**  **Horizontal Reflection**  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ROT(rows-i, cols-j) 1.jpg |
| ROWS-row, col |  | ROWS-row, COLS-col |
|  | | |
| Rotation is a linear image operation that reflects thepixels with respect to image rows and colums. It done by altering pixel row and column of index values. It is useful to rotate image. | | |

|  |  |  |
| --- | --- | --- |
| Zoom Image Pixels | | |
| TOP-LEFT | MIDDLE-CENTER | TOP-RIGHT |
| C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ZOOM TopLeft 1.jpg | C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ZOOM CenterCenter 1.jpg | C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ZOOM TopRight 1.jpg |
| BUTTOM-LEFT | ORIGINAL | BUTTOM-RIGHT |
|  |  |  |
| Zoom image operation focuses a frame of same size as image to portion of the image pixels, mathematical, pixels are selected repeatedly until it fills a frame. Useful of zooming images. | | |

|  |  |  |
| --- | --- | --- |
| Transform Image Pixels | | |
| RGB | GRB | BRG |
| C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\RGB 1.jpg  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\RBG 1.jpg | C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GRB 1.jpg  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GBR 1.jpg | C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\BRG 1.jpg  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\BGR 1.jpg |
| RBG | GBR | BGR |
|  |  |  |
| Pixel Transfrom is awesome. It swaps pixels values without altering the values of the image. There’re six possible arrangement of pixels R,G,B. It is useful for real time camera effects. | | |

|  |  |  |
| --- | --- | --- |
| Bitwise OR A = B|k ; k=0,1,2,…,254 | | |
| Frame=000 (RGB) | Frame=063 (RGB) | Frame=64 (GRB) |
| C:\Users\tenet\Desktop\Results\OR\OR 0 1.jpg |  | **C:\Users\tenet\SkyDrive\VisualStudioSpace2015\OpenCVWorkspace\Tutorial1\Tutorial1\Results\OR\OR 64 1.jpg**  **C:\Users\tenet\Desktop\Results\OR\OR 254 1.jpg** |
| Frame=128 (GBR) | Frame=192 (UNCLEAR) | Frame=254 (WHITE) |
|  |  |  |
| Bitwise OR is a logic image operation that makes brighter and high constract gradually as addition and multiplication operation. Every 64 frames the image non uniform transform R,G,B arrangement, moved from RGB 🡪 GRB 🡪 GBR 🡪 UNCLEAR🡪 WHITE.  (CONSTRAST , BRIGHTNESS, NON-UNIFORM TRANSFORM) | | |

|  |  |  |
| --- | --- | --- |
| Bitwise AND A = B&k ; k=0,1,2,…,254 | | |
| Frame=000 (BLACK) | Frame=063 (UNCLEAR) | Frame=064 (GBR) |
| C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\AND 0 1.jpg  C:\Users\tenet\Desktop\Results\AND\AND 128 1.jpg | C:\Users\tenet\Desktop\Results\AND\AND 63 1.jpg | C:\Users\tenet\Desktop\Results\AND\AND 64 1.jpg  C:\Users\tenet\Desktop\Results\AND\AND 254 1.jpg |
| Frame=128 (UNCLEAR) | Frame=192 (RGB) normal | Frame=254 (RGB) bright |
|  |  |  |
| Bitwise AND is a logic image operation that makes brighter and high constract gradually as addition and multiplication operation. Every 64 frames the image non uniform transform R,G,B arrangement, moved from BLACK 🡪 UNCLEAR🡪 GBR 🡪 UNCLEAR 🡪 RGB 🡪 RGB. Notice that OR operation continues effects of AND operation.  (CONSTRAST , BRIGHTNESS, NON-UNIFORM TRANSFORM) | | |

|  |  |  |
| --- | --- | --- |
| Bitwise XOR A = B|k ; k=0,1,2,…,254 | | |
| Frame=000 (RGB) | Frame=063 (RGB) | Frame=064 (GRB) |
| C:\Users\tenet\Desktop\Results\XOR\XOR 0 1.jpg  C:\Users\tenet\Desktop\Results\XOR\XOR 128 1.jpg | C:\Users\tenet\Desktop\Results\XOR\XOR 63 1.jpg  C:\Users\tenet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\XOR 192 1.jpg | C:\Users\tenet\Desktop\Results\XOR\XOR 64 1.jpg  C:\Users\tenet\Desktop\Results\XOR\XOR 254 1.jpg |
| Frame=128 (GBR) | Frame=192 (UNCLEAR) | Frame=254 (BGR) |
|  |  |  |
| Bitwise XOR is a logic image operation that makes radiantes surface of image as like remaider operation. Every 64 frames the image non uniform transform R,G,B arrangement, moved from RGB 🡪 GRB 🡪 GBR 🡪 UNCLEAR 🡪 BGR. No change to brightness and constrast of image.  (NON-UNIFORM TRANSFORM) | | |

|  |  |  |
| --- | --- | --- |
| Remainder REM A = B%k ; k=1,2,…,255 | | |
| Frame=001 | Frame=032 | Frame=064 |
| C:\Users\tenet\SkyDrive\VisualStudioSpace2015\OpenCVWorkspace\Tutorial1\Tutorial1\Results\REM\REM 1 1.jpg  C:\Users\tenet\Desktop\Results\REM\REM 128 1.jpg | C:\Users\tenet\Desktop\Results\REM\REM 32 1.jpg  C:\Users\tenet\Desktop\Results\REM\REM 192 1.jpg | C:\Users\tenet\Desktop\Results\REM\REM 64 1.jpg  C:\Users\tenet\Desktop\Results\REM\REM 254 1.jpg |
| Frame=128 | Frame=192 | Frame=254 |
|  |  |  |
| Remainder division is a non linear image operation, it increases the constrast of image while it transforms pixels differently within the same frame. It is obvious that transform is non uniform and non periodic. But constrast is incremental as addition.  (NON-UNIFORM TRANSFORM, CONSTRAST) | | |

|  |  |  |
| --- | --- | --- |
| Left Shift A = B<<k ; k=0,1,2,…,255 | | |
| Frame=001 | Frame=002 | Frame=023 |
|  | C:\Users\tenet\Desktop\Results\LSHIFT\LSHIFT 2 1.jpg | C:\Users\tenet\Desktop\Results\LSHIFT\LSHIFT 23 1.jpg  C:\Users\tenet\Desktop\Results\LSHIFT\LSHIFT 192 1.jpg |
| Frame=024 | Frame=031 | Frame=032 |
|  |  |  |
| Bitwise left shift is logic image operation, in every 32 frames, first 24 frames increases brightness of images as multiplecation and last 8 frames non-uniform transform image pixels within same frame. Exactaly the same effects repeat every 32 steps. Pixel color has 8 bits with RGB total of 24 bits.  (NON-UNIFORM TRANSFORM, BRIGHTNESS) | | |

|  |  |  |
| --- | --- | --- |
| Right Shift A = B>>k ; k=0,1,2,…,255 | | |
| Frame=001 | Frame=002 | Frame=023 |
|  |  |  |
| Frame=024 | Frame=031 | Frame=032 |
|  |  |  |
| Bitwise left shift is logic image operation, in every 32 frames, first 8 frames deccreases brightness of images as division and last 24 frames are black. First 24 steps make sense, since each pixel color has 8 bits with RGB total of 24 bits.  (BRIGHTNESS) | | |