Tehnici de imbunatatire si restaurare a imaginilor

Imbunatatirea unei imagini (Image enhancement) - cresterea calitatii imaginii la un nivel care usureaza interpretarea sa sau extragerea caracteristicilor existente in imagine. Operatii tipice de imbunatatire a imaginilor:

- Conversia nivelelor de gri
- Conversia histogramei
- Conversia culorilor
- Netezirea zgomotului
- Evidentierea caracteristicilor

Restaurarea unei imagini (Image restoration) – reconstruirea unei imagini degradate prin defectele introduse de sistemul de achizitie, cunoscand natura defectelor. Cauzele degradarii pot fi:

- Defecte ale lentilelor optice
- Neliniaritati ale senzorului (care transforma semnalul optic in semnal electric)
- Miscarea relativa dintre obiect si camera
- Focalizare gresita
- Turbulenta atmosferica
- etc.

Unele degradari sunt de natura geometrica altele se manifesta prin aparitia unor intensitati (culori) in imagine care nu redau puncte ale imaginii reale (zgomote).

Operatiile tipice de restaurare sunt:

- Corectia distorsiunilor geometrice
- Reducerea zgomotului din imagine

O imagine este un semnal electric bidimensional.

Tehnicile de imbunatatire si restaurare folosesc transformari care se aplica pe reprezentarea imaginii:

- in domeniul spatial (x,y): asupra pixelilor imaginii
- in domeniul frecventei (amplitudinea si frecventa semnalelor care produc imaginea): asupra Transformatei Fourier a imaginii

Operatiile de imbunatatire sunt, in general, transformari in domeniul spatial. Pentru restaurarea imaginilor se folosesc in special transformari in domeniul frecventei.

In cele ce urmeaza consideram imagini in mai multe nivele de gri. Multe dintre operatiile aplicate acestor imagini se extind asupra imaginilor color.

I. Tehnici de imbunatatire si restaurare in domeniul spatial

O imagine in mai multe nivele de gri se reprezinta in domeniul spatial printr-o functie bidimensionala, f(x, y), unde:

- (x,y) este o adresa de pixel imagine
- f(x,y) este intensitatea (nivelul de gri al) pixelului

Vom considera imaginea reprezentata printr-o matrice bi-dimensionala, img, astfel incat img[y][x] = f(x,y).

Notam cu T transformarea aplicata unei imagini f(x,y) – imaginea de intrare, care se doreste a fi imbunatatita- si cu g(x,y) imaginea rezultata prin aplicarea transformarii: g(x,y) = T(f(x,y)).

Transformarea T este definita pe o vecinatate a pixelului (x,y), de forma unei ferestre dreptunghiulare centrata in pixelul (x,y). Vecinatea poate fi redusa la pixelul (x,y).

Notam cu: 0, 1, 2, 3, …., L-1, nivelele de gri din imagine, Lmax = L-1.

In exemplele de implementare a diferitelor operatii este utilizat limbajul C. Matricea imagine este de tipul "imagine", definit astfel:

```
typedef unsigned char ** imagine;
```

1. Conversia nivelelor de gri

Imbunatatirea contrastului (Contrast stretching)

Fiind date 2 nivele de intensitate, Lval si Hval, Lval < Hval, transformarea este:

$$g(x,y) = \begin{cases} 0 & \text{daca } f(x,y) <= Lval \\ Lmax & \text{daca } f(x,y) >= Hval \\ s * (f(x,y) - Lval) & Lval < f(x,y) < Hval \end{cases}$$

unde $s = Lmax/(Hval - Lval)$, deci, intensitatile pixelilor $Lval < f(x,y) < Hval$ sunt scalate a.i. sa fie uniform distribuite in imagine.

Aplicarea unui prag (thresholding):

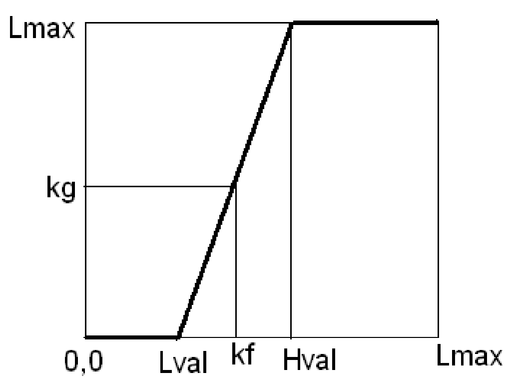
$$g(x,y) = \begin{cases} 0 & \text{daca } f(x,y) < P \\ Lmax & \text{daca } f(x,y) >= P \end{cases}$$

unde P este numita „valoare de prag“.

Aceste doua transformari pot fi definite si printr-o functie de transformare a nivelului de gri:

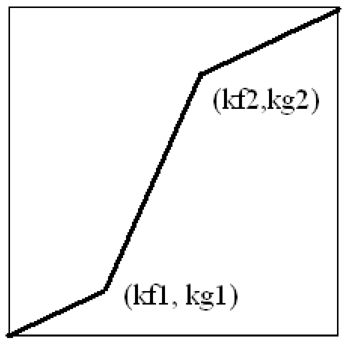
$$T: 0 .. (L-1) \rightarrow 0 .. (L-1)$$

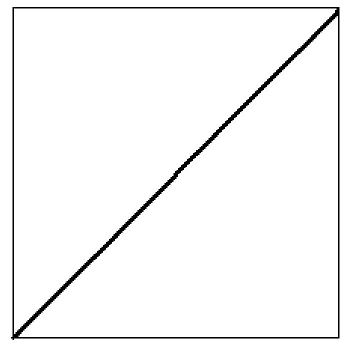
Domeniul de definitie reprezinta nivelele de gri din imaginea de intrare iar domeniul rezultatului nivelele de gri din imaginea de iesire.



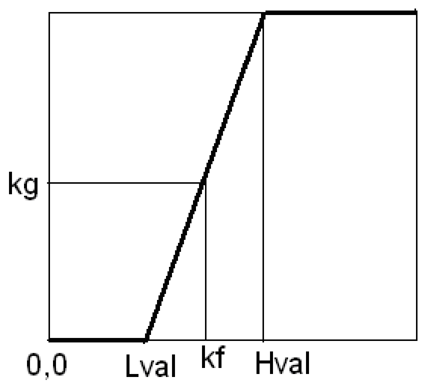
$$kg = T(kf) = \begin{cases} 0, & kf <= Lval \\ Lmax & kf >= Hval \\ s(Kf - Lval), & Lval < kf < Hval \end{cases}$$

In acest caz, T este o functie liniara definita pe portiuni. Cazul general in care T este o functie liniara definita pe portiuni poate fi reprezentat astfel:

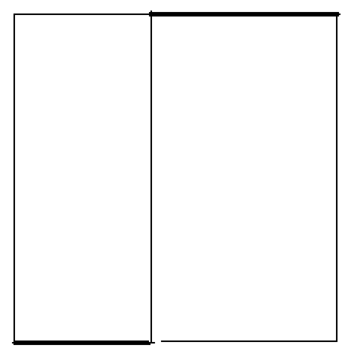




kf1=kg1, kf2=kg2
Nu se produc schimbari de intensitate



kf1=Lval, kg1 =0, kf2=Hval, kg2=Lmax



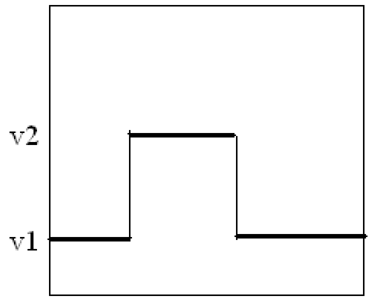
kf1=kf2, kg1=0, kg2=Lmax

Evidentierea caracteristicilor

Se evidentiaza un domeniu specific de nivele de gri din imagine, reducand totodata nivelele din afara domeniului la un nivel constant:

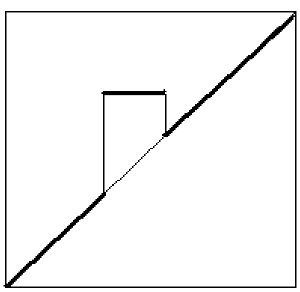
$$g(x, y) = \begin{cases} v1 & \text{daca } f(x,y) <= Lval \text{ sau } f(x,y) >= Hval \\ v2 & \text{altfel} \end{cases}$$

unde, in general, v1 este mic iar v2 este mare.



Se evidentiaza un domeniu specific de nivele de gri din imagine, lasand nemodificate nivelele din afara domeniului:

$$g(x, y) = \begin{cases} f(x,y) & \text{daca } f(x,y) <= Lval \text{ sau } f(x,y) >= Hval \\ v2 & \text{altfel} \end{cases}$$



Corectia Gamma

Pentru majoritatea monitoarelor, curba de raspuns intensitate – putere semnal (voltaj) este o functie, $I = voltaj^{2.5}$. Deci, daca cerem afisarea unui pixel in intensitatea k, pixelul va fi afisat cu intensitatea $k^{2.5}$. Deoarece nivelele de voltaj

trimise monitorului sunt intre 0 si 1, intensitatea afisata va fi mai mica decat aceea dorita. Monitoarele au gama = 2.5
Corectia gamma consta in ridicarea la putere a semnalului de intrare cu (1/2.5 = 0.4).

Deci, pentru imaginile afisate pe un monitor corectia gama este:
$g(x,y) = f(x,y)^{0.4}$

Solutia nu este generala, deoarece anumite sisteme au corectia gama inclusa in hardware. Asfel sunt Macintosh si Silicon Graphics. Calculatoarele Sun si PC nu au corectia gama inclusa in mod standard dar anumite placi grafice instalate pe aceste calculatoare pot face aceasta corectie.

O corectie asemanatoare ar trebui facuta unei imagini care este afisata pe ecran si se doreste tiparirea sa la o imprimanta. Imaginea imprimata poate aparea mai luminata.
Valoarea lui gama depinde de echipament.

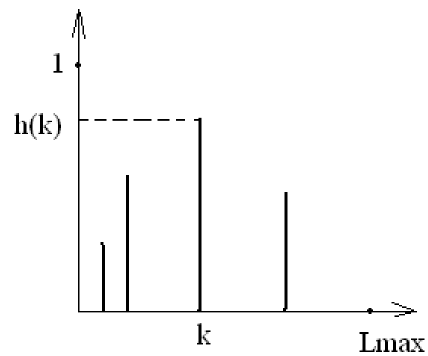
2. Prelucrari folosind histograma imaginii

Histograma unei imagini cu nivelele de gri k= 0, 1, ...L-1 este o functie discreta,

$h(k) = n_k$, unde n_k este numarul de pixeli din imagine, cu intensitatea k.

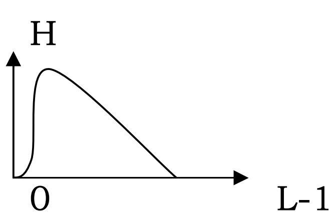
Histograma normalizata este definita prin functia:

$h(k) = n_k / n$, unde n este numarul total de pixeli din imagine.

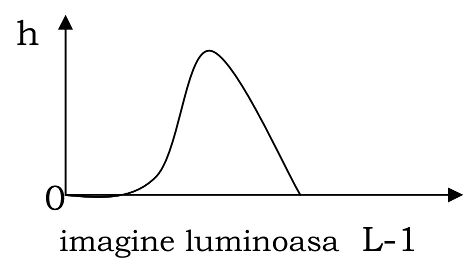


Histograma imaginii da informatii importante asupra continutului imaginii. Astfel:

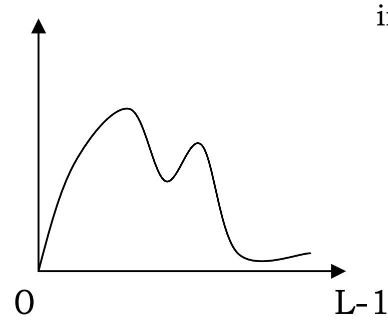
- daca valorile mari ale histogramei sunt concentrate in zona de intensitati mici, imaginea este intunecata.
- daca valorile mari ale histogramei sunt concentrate in zona de intensitati mari, imaginea este luminoasa.



Imagine intunecata

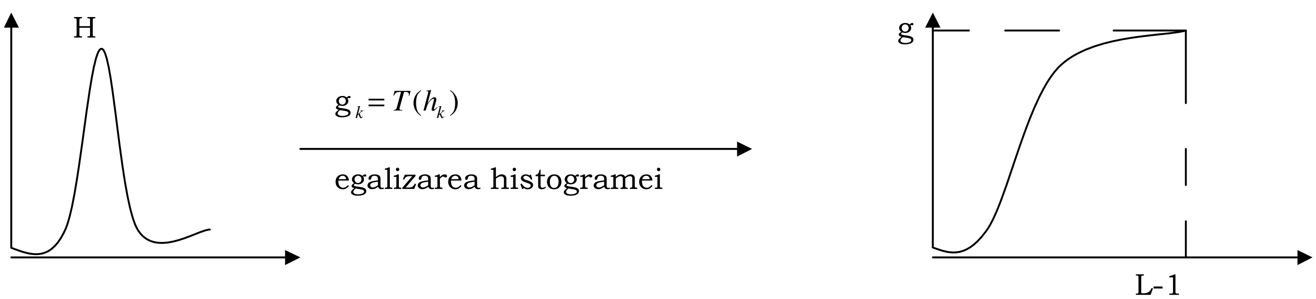


imagine luminoasa



- daca imaginea contine 2 obiecte cu intensitati diferite sau un obiect care se distinge clar de fond, histograma prezinta doi lobi (varfuri de maxim local, respectiv un varf de minim local).

Daca histograma este concentrata intr-o zona ingusta de intensitati, contrastul imaginii este slab si calitatea imaginii (in privinta perceperii) este slaba. Calitatea imaginii poate fi imbunatatita modificandu-i histograma, printr-o transformare numita **"egalizarea histogramei sau liniarizarea histogramei".**



Probabilitatea de aparitie in imagine a nivelului de intensitate k este aproximata prin:

$pr(k) = n_k / n$, k=0,..L-1

Transformarea de egalizare a histogramei este definita astfel:

$$g_k = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} h_j$$

Intensitatea k din imaginea de intrare se va inlocui cu intensitatea g_k * (L-1) in imaginea de iesire.

Calculul histogramei normalizate:

```
typedef unsigned char **imagine;

void histo(imagine a, int H, int W, int L, float *h)
{unsigned long*Ih,n;
 * alocare Ih
 for (i=0;i<L;i++) Ih[i]=0;
 for (i=0; i<H; I++)
         for (j=0; j<W;j++)
            Ih[ (unsigned int) a[i][j]]++;
 n=(float) H*W;
 for (i=0; i<L; i++)
         h[i]=Ih[i]/n;
 * dealocare Ih;
}
```

Functia pentru calculul imaginii pe baza histogramei transformate:

```
void histEgal (imagine In, imagine Out, int H, int W, int L)
{ int i, j;
  int *f = (   ) malloc(  );
  float * h = (  ) malloc(L*sizeof(   ));
  float *g = …….
// Calcul histograma imagine de intrare
 histo( In, H, W, L, h);
// transforma histograma
  g[0] = h[0];
  for ( I=1; I<L; I++ )
         g[i] = g[i-1] + h[i]
// Calculeaza intensitatile din imaginea de iesire
for(i = 0; i<L; ++ )
```

```
    f[i] = g[i]* (L-1);
 for(i = 0; i<H; I++ )
    for(j = 0; j<W; j++ )
         Out[i][j] = ((unsigned char) f[(unsigned int)In[i][j]];
* dealocare f, h, g
}
```

Imaginea de iesire are un contrast mult mai bun decat imaginea de intrare deoarece intensitatile din imaginea de intrare sunt scalate a.i. sa fie relativ uniform distribuite in imagine. Egalizarea histogramei tinde sa amplifice zgomotul.

Alte prelucrari pe baza de histograma au drept scop:

- reducerea numarului de nivele de gri din imaginea initiala
- transforma histogramei intr-una specificata.

3. Corectia distorsiunilor geometrice

Corectia se bazeaza pe cunoasterea modelului matematic al distorsiunilor. Astfel, distorsiunea geometrica este definita printr-un set de ecuatii de transformare a pozitiilor pixelilor, din pozitiile reale (x,y) in pozitiile distorsionate (x',y'):

$$x' = h_x(x,y)$$
$$y' = h_y(x,y)$$

h_x si h_y sunt de obicei liniare pentru distorsiunile perspectiva sau cuadrice pentru distorsiunile produse de camere video.

Fie f(x,y) imaginea distorsionata si g(x,y) imaginea corectata. Corectia poate fi descrisa astfel:

Pentru fiecare pixel (x, y) al imaginii f(x,y)

Calculeaza $x' = h_x(x,y)$ si $y' = h_y(x,y)$

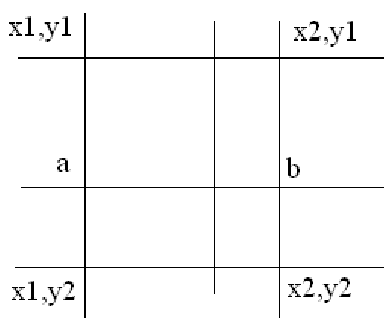
$g(x,y) = f(x', y')$

(x',y') poate sa nu fie o adresa de pixel – x', y' sunt numere reale. In acest caz valoarea g(x,y) se calculeaza prin interpolarea valorilor pixelilor intre care se afla pozitia (x',y'). De exemplu:

$x1 < x' < x2$, $y1 < y' < y2$

g(x,y) = f(x',y') se va aproxima astfel:

$$a = (y2 - y') * g(x1,y1) + (y'-y1) * g(x1,y2)$$
$$b = (y2 - y') * g(x2,y1) + (y'-y1) * g(x2,y2)$$
$$g(x,y) = (x2-x') * a + (x'-x1) * b$$



4. Reducerea zgomotului ("noise") din imagini

Sistemele optice si electronice folosite la producerea imaginilor introduc zgomote a caror intensitate depinde de calitatea aparaturii. Zgomotul este o informatie nedorita care deterioreaza calitatea unei imagini. Reducerea zgomotului are ca scop imbunatatirea calitatii imaginii.

4.1. Estimarea zgomotului dintr-o imagine

Fie n imagini ale aceleiasi scene, obtinuta cu acelasi sistem de achizitie si in aceleasi conditii de achizitie:

E_0, ... E_{n-1} de N*N pixeli

Pentru fiecare pixel (i,j) se calculeaza, **media:**

$$\overline{E}(i,j) = \frac{1}{n}\sum_{k=0}^{n-1} E_k(i,j)$$

$$\sigma(i,j) = \left(\frac{1}{n-1}\sum_{k=0}^{n-1}\left(\overline{E}(i,j) - E(i,j)\right)^2\right)^{\frac{1}{2}}$$

σ(i,j) este o estimare a **deviatiei standard a zgomotului de achizitie in pixelul (i,j)**;

$$\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sigma(i,j)}{N \times N}$$

media zgomotului pentru intreaga imagine,

$$\max_{\substack{0 \le i \le N-1 \\ 0 \le j \le N-1}}\left(\sigma(i,j)\right)$$

zgomotul maxim.

Camera video introduce si un alt tip de zgomot: valorile pixelilor nu sunt complet independente una de alta, fotosenzorii de pe acelasi rand nu produc valori independente. Se poate verifica folosind autocovarianti C_{EE} (i,j) imaginii unui sablon uniform spatial paralel cu planul imaginii si iluminat de lumina difuza:

$$C_{EE}(i',j') = c \cdot \sum_{i=0}^{N_{i'}} \sum_{j=0}^{N_{j'}} \left(E(i,j) - \overline{E}(i,j)\right)\left(E(i+i',j+j') - \overline{E}(i+i',j+j')\right)$$

$$c = \frac{1}{N^2} \quad , \quad N_{i'} = N - i' - 1 \quad , \quad N_{j'} = N - j' - 1$$

4.2. Modelarea zgomotelor

Zgomotul din imaginile digitale poate proveni dintr-o multitudine de surse. Procesul de achizitie al imaginilor digitale, care converteste o imagine optica intr-un semnal electric continuu este un proces primar generator de zgomote.

La fiecare pas din procesul de achizitie exista fluctuatii cauzate de fenomene naturale si acestea aduaga o valoare aleatoare la extragerea fiecarei valori a luminozitatii pentru un pixel dat.

Exista doua tipuri de zgomote:

- Independente de continutul imaginii
- Dependente de continutul imaginii

O imagine cu zgomot independent de continutul imaginii poate fi modelata prin:

g(x,y) = f(x,y) + n(x,y)

unde:

f(x,y) este imaginea de intrare pentru dispozitivul de formare a imaginii (imaginea reala) n(x,y) reprezinta zgomotul independent de continutul imaginii, numit si ***zgomot aditiv***

In cazul in care zgomotul depinde de continutul imaginii (de exemplu, radiatii monocromatice produse de o suprafata, care produc interferente de unde), zgomotul poate fi reprezentat printr-un model ne-liniar. Deoarece aceste modele matematice sunt mai complicate, zgomotul este considerat, daca este posibil, ca fiind independent de date (continutul imaginii).

Modelarea matematica a zgomotelor este utila nu numai pentru reducerea lor ci si pentru sinteza unor imagini cu zgomote tipice, in scopul analizei algoritmilor de filtrare a zgomotelor.

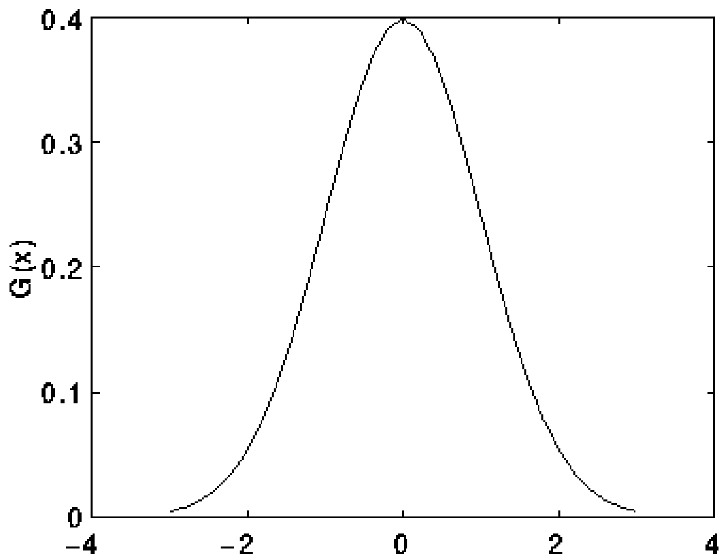
Astfel, tipurile uzuale de zgomot sunt: zgomotul **gaussian,** zgomotul **uniform** si cel de tip **salt-and peper (sare si piper** / **impuls).** Ele sunt modelate prin caracteristicile lor de probabilitate.

Zgomotul gaussian

Distributia Gaussiana 1-D are forma:

$$\mathbf{G(x) = (1\ /(2\pi\sigma))^{1/2}\ \ exp(\ -(x-m)^2/\ 2\ \sigma^2\ )}$$

unde x este o variabila aleatoare, m este media distributiei iar σ reprezinta deviatia standard a distributiei. Figura urmatoare reda distributia Gaussiana 1-D cu media zero (centrata in x = 0) si $\sigma = 1$:



Zgomotul Gaussian este modelat prin functia de densitate a probabilitatii,

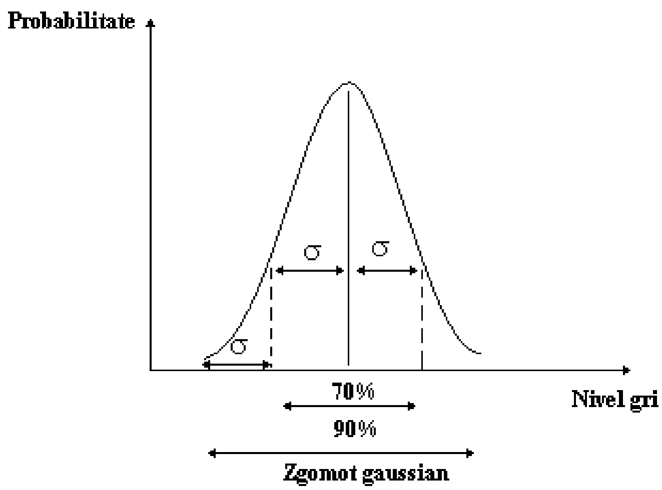
$$\mathbf{FDP(g) = (1 /(2\pi\sigma)^{1/2})\ exp(\ -(g\text{-}m)^2/ 2\ \sigma^2\ )}$$

unde:

g este nivelul de gri;

m = media zgomotului;

σ = deviatia standard a zgomotului;



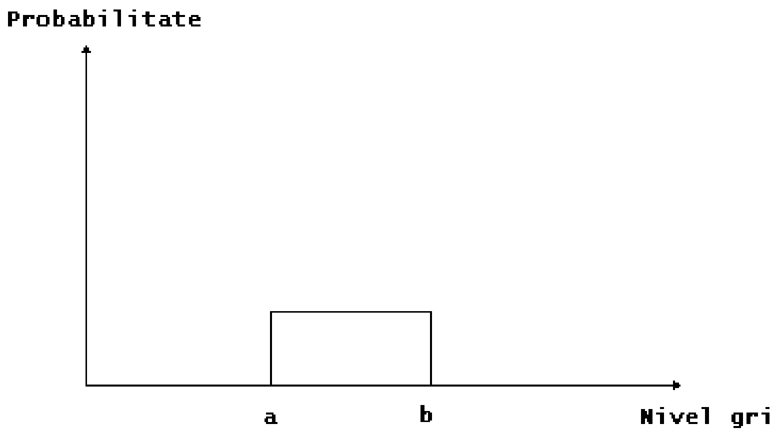
Aproximativ 70% din valori sunt incadrate intre medie ± σ (deviatia standard) si 95% dintre valori sunt cuprinse intre medie ± 2σ. Se poate considera ca valoarea acestei FDP este nula la aproximativ 3σ de valoarea medie.

Fiecare pixel din imaginea cu zgomot Gaussian are o valoare rezultata din insumarea valorii reale a pixelului cu o valoare aleatoare cu distributie Gaussiana.

n(x,y) = FDP(f(x,y))

Modelul gausian este cel mai des folosit pentru modelarea proceselor cu zgomot natural, cum ar fi cele care provin din zgomotul electronic in sistemul de achizitie al imaginilor.

Zgomotul uniform

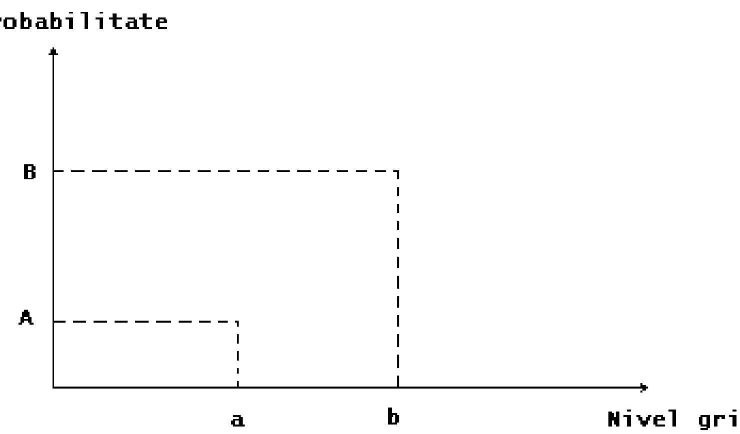


Functia de distributie este:

FDP(g) = 1/(b-a), a<=g<=b

FDP(g) = 0, altfel

Zgomot sare si piper (salt-and-peper)



Functia de distributie este:

FDP(g) =A, pentru g=a

FDP(g) =B, pentru g=b

FDP(g) =0, altfel

In modelul de zgomot de tip *salt-and-peper* exista doar doua valori posibile, A si B. Din aceasta cauza se mai numeste si zgomot de tip impuls (speckle). Probabilitatea de aparitie a fiecaruia este mai mica de 0.1; la valori mai mari decat acestea, zgomotul va domina imaginea. Pentru o imagine de 8 biti, valoarea de intensitate tipica pentru *zgomotul peper* este 0 si pentru *zgomotul salt* este 255.

Zgomotul salt-and-peper este in general cauzat de functionarea proasta a celulelor din senzorii camerelor, greseli ale locatiilor de memorie, erori de sincronizare in procesul de digitizare sau erori (pierderi de biti) pe canalul de comunicatie in cazul transmisilor imaginilor.

Alte tipuri de zgomote sunt cele avand FDP de urmatoarele tipuri:
- zgomot exponential negativ
- zgomot gamma
- zgomot Rayleigh

Nota: a se consulta fisierul “image processing.zip” de pe site.

4.3. Filtre spatiale

Filtrarea spatiala a imaginilor digitale este o operatie care se aplica local, la nivelul fiecarui pixel din imagine, inlocuind valoarea pixelului curent cu o valoare ce depinde de valorile pixelilor vecini (de aici si denumirea lor de filtre spatiale – se considera o vecinatate spatiala in sistemul de coordonate asociat imaginii). Aceste filtre opereaza pe vecinatati mici, intre 3x3 si 11x11.

Principalele categorii de filtre spatiale folosite la inlaturarea zgomotelor sunt *filtrele de mediere* si *filtrele ordonate(filtre rang)*.

Filtrele de mediere sunt in general filtre liniare aplicate printr-o operatie de convolutie a imaginii cu un nucleu (masca) de convolutie. Ele determina o mediere ponderata a vecinilor. Sunt mai eficiente pentru imagini cu *zgomot uniform* sau *zgomot gaussian*.

Filtrele ordonate sunt filtre neliniare implementate prin ordonarea crescatoare a valorilor pixelilor din vecinatatea pixelului curent. Aceasta ordonare se foloseste pentru a selecta una dintre valori. *Filtrele ordonate* (in special *filtrul median*) sunt mai eficiente pentru imaginile cu *zgomote salt-and-peper, zgomote exponential negative* si *zgomote Reyleigh.*

Un filtru care isi schimba comportamentul bazandu-se pe caracteristicile nivelelor de gri ale vecinilor este numit *filtru adaptativ,* si aceste filtre sunt des folosite in multe aplicatii practice.

4. 3.1. Filtre de mediere

Filtrele de mediere sunt filtre spatiale liniare. Ele modifica valoarea fiecarui pixel calculand o medie ponderata a valorilor pixelilor dintr-o vecinatate a sa (**fereastra de filtrare)**. Ponderile sunt definite intr-o matrice de forma vecinatatii, numita ***masca de convolutie*** sau **kernel**.

Aplicarea filtrului peste intreaga imagine este numita adesea ***convolutia imaginii*** cu ***masca*** de convolutie.

Fie f(x,y) imaginea pe care se aplica filtrul si w masca de convolutie, un dreptunghi cu laturile de 2*a, 2*b. Imaginea filtrata in pixelul (x,y) este data de convolutia discreta:

$$\mathbf{g(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i,j) * f(x+i, y+j)}$$

Diferitele tipuri de filtre de mediere se deosebesc prin valorile w(i,j).

Daca toate valorile din w(I,j) sunt pozitive, filtrul realizeaza o netezire medie (average smoothing) a imaginii.

Filtrul medie

Cel mai simplu filtru de mediere este ***filtrul medie***, care inlocuieste valoarea fiecarui pixel cu media aritmetica a valorilor pixelilor din fereastra de filtrare. De exemplu, pentru a=1, b=1 (fereastra de filtrare 3x3 pixeli), filtrul medie este definit astfel:

$$\mathbf{g(x,y) = 1/9 \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j)}$$

Filtrul medie este un filtru de netezire (**FTJ- filtru “trece jos**”). Tot un FTJ este si filtrul cu urmatoarea masca:

W(0,0) = 1/4

W(1,0) = w(0,1) = w(0,-1) = w(-1,0) = 1/8

W(1,1) = w(1,-1) = w(-1,-1) = w(-1,1) = 1/16

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Un FTJ atenueaza componentele de inalta frecventa din imagine, care pot reprezenta zgomote. Imaginea rezultata din aplicarea unui FTJ este mai incetostata (neclara) decat imaginea originala. Filtrul atenueaza tranzitiile bruste de intensitate lasand impresia ca imaginea are mai putine detalii.

Un filtru trece sus (FTS) accentueaza componentele de inalta frecventa, avand un efect mic asupra celor de joasa frecventa. In imaginea rezultata din aplicarea unui FTS sunt accentuate diferentele de intensitate (detaliile) in zonele de tranzitie de intensitate.

Un exemplu de FTS este filtrul cu urmatoarea masca:

W(0,0) = 9

w(i,j) = -1, pentru toate celelalte elemente

De asemenea, filtrul Laplacian cu masca:

W(0,0) = 5

w(-1,0) = W(1,0) = w(0,-1) = w(0,1) = -1

W(1,1) = w(1,-1) = w(-1,-1) = w(-1,1) = 0

0	-1	0
-1	5	-1
0	-1	0

Functia prezentata in continuare aplica filtrul medie pe un dreptunghi din imaginea a, reprezentat prin colturile sale (N1, M1), (N2, M2), rezultatul fiind memorat in matricea b. Masca de filtrare are w x h pixeli.

```
void f_net (imagine a, imagine b, int w, int h, int N1, int M1, int
N2, int M2)
{
  float wh, s;

  wh= (float)(w*h);

  w2 = w/2; h2 = h/2;

  for ( k = N1 + w2;  k < N2 – w2;  k++)
        for ( l = M1 + h2;  l < M2 – h2;  l++)
```

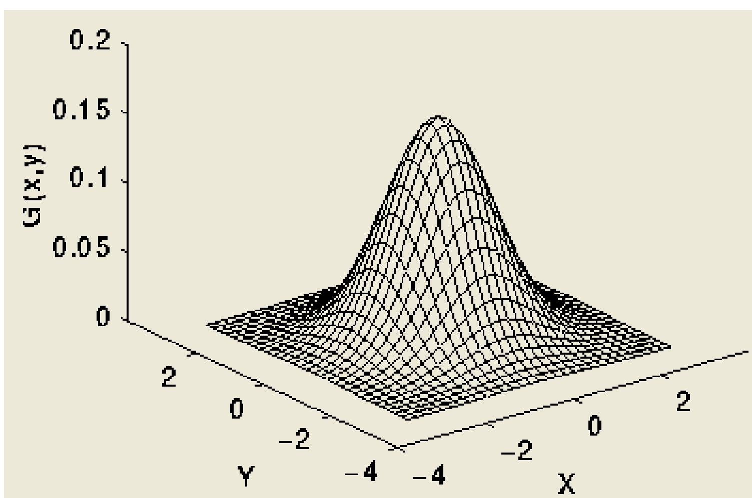
```
        {
                s=0;
                for ( i = -h2;  i <= h2;  i++)
                        for ( j = -w2;  j < w2;  j++)
                                s + = a[l+i][k+j];
                b[k][l] = (unsigned char) (s/wh);
        }
}
```

Filtrul gaussian

Functia de distributie Gaussiana 2D, cu m=0, are forma:

$$g(x,y) = (1/(2\pi\sigma^2))\, \exp(-(x^2 + y^2)/(2\sigma^2))$$

Figura urmatoare ilustreaza distributia cu media (0,0) si $\sigma = 1$:



Ideea netezirii Gaussiene este de a folosi aceasta distributie spatiala ca o functie de “acoperire” a pixelilor imaginii si aceasta se realizeaza prin convolutie. Masca filtrului este o aproximare discreta a acestei functii. Cu toate ca teoretic functia este diferita de zero in orice punct, in practica ea poate fi considerata nula la o distanta de aproximativ 3σ fata de medie. Deci, masca de convolutie poate fi limitata la aceasta valoare. Urmatoarea masca de convolutie cu valori intregi aproximeaza distributia Gaussiana cu deviatia standard (σ) = 1.0

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

O alta masca de convolutie, care aproximeaza distributia Gaussana cu σ = 1.4, este:

$\frac{1}{115}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Filtrul Gaussian este un filtru de netezire, ca si filtrul medie. Gradul de netezire este determinat de valoarea deviatiei standard. Cu cat valoarea deviatiei este mai mare cu atat nucleul (masca) de convolutie este mai mare. Netezirea este cu atat mai puternica cu cat valoarea σ si dimensiunea nucleului de convolutie gaussian sunt mai mari. In general valoarea σ a filtrului trebuie corelata cu nivelul zgomotului din imagine: un zgomot gaussian puternic se va filtra cu un nucleu gaussian si σ mai mari.

Filtrul Gaussian se deosebeste de filtrul medie (care acorda ponderi egale pixelilor din fereastra de filtrare) prin faptul ca efectueaza o medie ponderata a pixelilor din fereastra, cu ponderile crescatoare spre pixelul din centru. Din aceasta cauza, filtrul Gaussian produce o netezire mai fina si conserva frontierele mai bine decat un filtru medie de aceeasi dimensiune.

Filtrul gaussian poate fi folosit pentru eliminarea *zgomotului gaussian* si a *zgomotului uniform.*

O masca de 3x3 care aproximeaza convolutia Gaussiana este urmatoarea:

$$M(x, y) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

4. 3. 2. Filtre ordonate (filtre rang)

In general, zgomotele din imagine apar in domeniul frecventelor inalte din spectrul imaginii. De aceea, pentru inlaturarea lor se folosesc filtre trece-jos (filtre de netezire). Filtrele liniare trece-jos tind sa reduca detaliile din imagini (frontiere, colturi), care sunt reprezentate prin frecvente inalte. Ele diminueaza punctele de front, degradand in acest fel imaginea.

Filtrele neliniare reduc zgomotele, pastrand totodata fronturile si detaliile din imagine.

O clasa de filtre neliniare este aceea a filtrelor ordonate, numite si filtre rang. Ele se bazeaza pe o ordonare a intensitatilor pixelilor din fereastra de filtrare.

Fie P=(x,y) un punct al imaginii si Vp= rxs o vecinatate a sa. Fie $f_1, f_2, .., f_N$ intensitatile pixelilor din vecinatatea Vp si R_{Vp} ordonarea lor crescatoare (vectorul ordonat al intensitatlor)

$$R_{Vp} = \{ f_1, f_2, \ldots, f_N \}, \qquad f_i \leq f_{i+1}$$

Aplicarea unui filtru rang este definita astfel:

$$g(p) = \mathrm{Rang}_{\mathbf{j}} (R_{Vp})$$

adica, intensitatea pixelului P in imaginea filtrata va fi aceea memorata in pozitia j a vectorului R_{Vp}.

Pentru j = 1 se obtine *filtrul min:*

$$g(p) = \min (R_{Vp}) = \min \{ f(p) \mid p \in V\}$$

Pentru j = N se obtine *filtrul max:*

$$g(p) = \max (R_{Vp}) = \max \{ f(p) \mid p \in V\}$$

Filtrul median

Este definit pentru N impar. El corespunde pozitiei de mijloc in vectorul R, deci:

$$g(p) = \mathrm{Rang}_{\mathbf{(N+1)/2}} (R_{\mathbf{Vp}})$$

g (p) este intensitatea mediana dintre cele N intensitati.

Filtrul *max* poate elimina impulsurile negative (spoturi negre).

Filtrul *min* elimina impulsurile pozitive (spoturi albe).

Filtrul median inlatura zgomotele fara a degrada imaginea deci fara a atenua punctele de front. Filtrele liniare de netezire tind sa distribuie zgomotele in punctele inconjuratoare acelora care reprezinta zgomot.

Exemplu: N=5, R_{Vp} = { 100, 110, 120, 130, 250}.

Intensitatea 250 poate fi un zgomot (impuls) sau un punct caracteristic in imagine. Iesirea filtrului median va fi 120, in timp ce a filtrului medie 710/5 = 142.

Alegerea vecinatatii (fereastra de filtrare)

Forma vecinatatii in care se calculeaza cele N intensitati influenteaza efectul filtrului. In general, vecinatatea se alege mica pentru a se evita efortul de calcul presupus de sortarea unui vector mare de intensitati.

Alegerea formei ferestrei se poate baza pe o cunoastere a caracteristicilor de zgomot din imagine, de ex. orientarea orizontala sau verticala.

Fereastra de filtrare pentru filtrul median poate fi de forma patrata (ex. 3x3), dreptunghiulara sau cruce.

Imaginea poate fi filtrata de mai multe ori folosind aceeasi fereastra.

Proprietatile filtrului median

1. Reduce variatia intensitatilor din imagine, producand regiuni de intensitate constanta sau aproape constanta. Forma regiunilor depinde de geometria ferestrei de filtrare, aspect neplacut deoarece sunt introduse in imaginea filtrata regiuni care nu existau.
2. Netezeste oscilatiile de intensitate cu o perioada mai mica decat latimea ferestrei.

3. Modifica valoarea medie a intensitatilor din imagine daca distributia spatiala a zgomotului nu este simetrica in fereastra.
4. Conserva anumite forme de frontiere.
5. Nu genereaza noi nivele de gri.

Aplicarea filtrului median

La aplicarea filtrelor liniare se efectueaza calcule de inmultire si adunare. Aplicarea filtrului median presupune sortarea unui vector de numere intregi. Acest calcul poate fi eficientizat in mai multe moduri. Astfel, vectorul corespunzator pozitiei (x+1, y) se poate obtine prin modificarea vectorului sortat corespunzator pozitiei (x, y): se elimina din vector pixelii de pe prima coloana a ferestrei, ramanand (n x m – 2n) pixeli din vechiul vector.

Algoritmul Huang (pentru aplicarea filtrului median)

Fereastra de NxM pixeli este deplasata pe randuri, de la stanga la dreapta. Ferestrele centrate in pixelii (i, j) si (i+1, j) au in comun (M x N - 2N) pixeli.

Calculul valorii mediane din fereastra curenta se bazeaza pe valoarea mediana a ferestrei anterioare si pe histograma imaginii din fereastra.

Astfel, valoarea mediana poate fi determinata stiind ca N/2 pixeli ai imaginii din fereastra au intensitati mai mici decat ea.

Ex: 4, 6, 8, 10, 11, 12, 14 (N=7)

Valoarea mediana este 10; pozitia (N+1)/2 = 4

In vectorul sortat al intensitatilor, N/2 intensitati sunt mai mici decat valoarea mediana.

Daca valoarea mediana corespunde pozitiei i in histograma (nivelul i de intensitate), atunci numarul total de pixeli cu intensitatea mai mica decat i este

$$\sum_{j=0}^{i-1} h(j)$$

unde h este vectorul histograma (l(k)=numarul de pixeli cu intensitatea k).

Algoritmul Huang in pseudocod:

Pentru fiecare rand al imaginii
- *Pentru prima* fereastra de pe rand
  - calculeaza histograma
  - construieste si sorteaza vectorul pixelilor din fereastra
  - determina valoarea mediana
- *Pentru* celelalte ferestre de pe rand
  - actualizeaza histograma:
    - scazand contributiile pixelilor de pe coloana din stanga a ferestrei
    - adaugand contributiile pixelilor de pe coloana din dreapta a ferestrei
  - determina valoarea mediana a ferestrei folosind valoarea mediana a ferestrei anterioare, care se ajusteaza impunand conditia: (nr.pixeli cu intensitate < med) = N/2

Implementarea in C a algoritmului:

```
void Huang-median (imagine a, imagine b,
                    int N, int M,    //laturile ferestrei
                    int Nmax, int Mmax )//dim. Imagine
        {
        int k, l, i, j, NM, *t;
        unsigned long hist[256];
        t = (int*) malloc ( (unsigned int) (N*M) * sizeof (int) );
```

```
NM = N*M;
NM2 = NM/2;
N2 = N/2;
M2 = M/2;
for ( k = N2; k < Nmax - N2; k++ )
        {
        /*initializarea histogramei*/
        for ( i = 0; i < 256; i++)
                hist[i]=0;
        // calc. histograma si val. mediana pt. prima fereastra
        for ( i = 0; i < N; i++ )
                for ( j = 0; j < M; j++ )
                        {
                        g = t [M * i + j] = a[k + i – N2 ][j];
                        hist [g]++;
                        }
                sortare (t, NM);
                med = t [NM2];  //val mediana
                b[k][M2] = med;
        //calculeaza nr. de pixeli cu intensitatea < med
        ltmed = 0;
        for ( i = 0; i < med; i++)
                ltmed+=hist[i];
        //calculeaza val mediana pt celelalte ferestre de pe linia k
        for ( l = M2+1; l < Mmax – M2; l++)
                {
                 //calculeaza histograma si ltmed
                 for ( i =0; i < N; i++)
                        {
                        g = a [k+i-N2][l-M2-1];
                        hist [g]--;
                        if (g < med)   ltmed--;
                        g = a [k+i-N2][l+M2];
                        hist [g] ++;
                        if (g<med)     ltmed++;
                        }

                //calculeaza valoarea mediana a ferestrei l
                if (ltmed > NM2)
```

```
                        do{
                                med --;
                                ltmed- = hist [med];
                                }
                        while (lmed > NM2);
                else
                        while (ltmed + hist [med] <= NM2)
                                {
                                        ltmed += hist [med];
                                        med ++;
                                        }
                b [k][l] = med;

                }       //for l

        }       //for k
free (t);
}
```

Algoritmul este mai rapid decat in cazul folosirii sortarii in fiecare fereastra, chiar daca se foloseste un algoritm de sortare rapid.

Observatie:

Filtrul nu se aplica pe marginea de latime N/2, repectiv M/2 a imaginii. Pentru filtrarea acestor margini se foloseste o fereastra trunchiata.

O alta metoda de reducere a timpului de calcul la aplicarea filtrului median consta in calculul iterativ al valorii mediane din R_{Vp} , bit cu bit, incepand cu bitul cel mai semnificativ.

Metoda este ilustrata de urmatorul exemplu:

Fie { 3,9,4,7,5} intensitatile pixelilor din fereastra.

1. Se determina bitul valoare medie de pe pozitia cea mai semnificativa . Acesta va fi bitul cel mai semnificativ al valorii mediane.

b3	b2	b1	b0
0	0	1	1
1	0	0	1
0	1	0	0
0	1	1	1
0	1	0	1

med = 0

2. Se continua numai cu valorile care au b3 = 0

   Se determina bitul valoare medie din pozitia b2

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

med = 1

3. Se procedeaza la fel pentru a obtine bitii b1 si b0 ai valorii mediane

1	0	0
1	1	1
1	0	1

med = 0

$\longrightarrow$

0	0
0	1

med = 1

Valoarea mediana ar putea fi 4 (alegand b0 = 0) sau 5 (b0 =1). Dar, valoarea mediana trebuie sa fie mai mare decat doua elemente ale vectorului (N= 5, N/2 = 2), deci se alege b0 =1.

Atunci cand pe o pozitie numarul de biti = 0 este egal cu numarul de biti = 1, sunt necesare teste suplimentare pentru alegerea valorii bitului median. Testul foloseste conditia ca N/2 elemente ale vectorului sa fie mai mici ca valoarea mediana.

II. Tehnici de imbunatatire si restaurare in domeniul frecventei

1. Transformata Fourier

Exista mai multe tipuri de transformari ale imaginilor care produc reprezentari in care apar proprietati ale imaginilor nedisponibile in spatiul imaginii. Astfel, transformata Fourier a unei imagini este o reprezentare in domeniul frecventei. Multe prelucrari de imagine presupun eliminarea din imagine a componentelor de o anumita frecventa, de exemplu cele de nivel coborat sau cele de nivel inalt. Aceste operatii sunt usor de realizat pe transformata Fourier a imaginii.

O prelucrare de imagine bazata pe transformata Fourier are loc in trei pasi:

- *Se calculeaza transformata Fourier a imaginii*
- *Se proceseaza reprezentarea in domeniul Fourier*
- *Se calculeaza transformata Fourier inversa, obtinandu-se imaginea prelucrata*

Transformata Fourier a unei imagini *I(x,y)* este definita astfel:

$$F(u,v)=\iint I(x,y)\, exp[-j2\pi(ux + vy)]dxdy$$

iar *inversa transformarii*:

$$I(x,y)=\iint F(u,v)\, exp[j2\pi(ux+vy)]dudv$$

Domeniul Fourier este complex si poate fi reprezentat ca:

$$F(u,v)=Real(u,v) + j\, Imag(u,v)$$

In prelucrarea imaginilor se folosesc *amplitudinea* si *faza*:

$|F(u,v)| = (Real(u,v)^2 + Imag(u,v)^2)^{1/2}$

$\Phi(u,v) = tan^{-1} (Imag(u,v) / Real(u,v))$

Fie *I(x)* un semnal 1D. Transformata Fourier a functiei *I(x)* este:

$$F(u) = \int_{-\infty}^{\infty} I(x)[\cos(2\pi ux) - j\sin(2\pi ux)]dx = \int_{-\infty}^{\infty} I(x)\exp(-j2\pi ux)dx$$

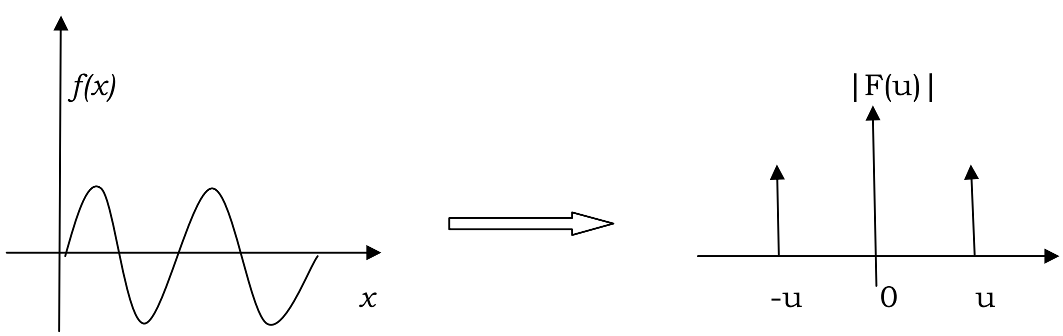
unde j= $\sqrt{-1}$ iar u reprezinta frecventa.

Semnalul *F(u)* poate fi transformat din domeniul frecventei in domeniul spatial prin transformata Fourier inversa:

$$I(x) = \int_{-\infty}^{\infty} F(u)[\cos(2\pi ux) + j\sin(2\pi ux)]du = \int_{-\infty}^{\infty} F(u)\exp(j2\pi ux)du$$

|*F(u)*| reprezinta spectrul amplitudinii semnalului.

I(x) se descompune intr-un set de componente de unda sinusoidale,f(x). Fiecare punct in |*F(u)*/ specifica amplitudinea si frecventa unei singure componente sinusoidale de unda:



O functie bi-dimensionala, I(x,y) (o functie imagine), se decompune intr-un set de frecvente spatiale F(u,v). Un punct F(u,v) este o masura a frecventei spatiale (u,v) in imagine.

O frecventa spatiala, F(u,v), este o suprafata cu ondulatii sinusoidale a caror rata de ondulatie este data de distanta punctului (u,v) fata de origine: $(u^2 + v^2)^{1/2}$

Deci, toate frecventele spatiale cu aceeasi rata de ondulatie se reprezinta prin puncte aflate pe circumferinta unui cerc: $r = (u^2 + v^2)^{1/2}$

Orientarea suprafetei (unghiul fata de axa OX) este data de $\tan^{-1}(u/v)$.

Un singur punct, F(u,v), ne spune cat de mult dintr-o anumita frecventa este continut intr-o imagine.

2. Transformata Fourier discreta (Discrete Fourier Transform)

Fie *I(x)*, $0 \le x \le N-1$, o secventa discreta reprezentand un semnal 1D esantionat. Transformata Fourier a semnalului astfel reprezentat se numeste *transformata Fourier discreta:*

$$F(u) = \sum_{0 \le x \le N-1} I(x) \exp(-j 2\pi u x / N)$$

O imagine, I(x,y), este un *semnal discret bi-dimensional*

$$0 \le x \le N_1 - 1, \; 0 \le y \le N_2 - 1$$

Transformata Fourier a imaginii I(x,y) este:

$$F(u,v) = \sum_{x=0}^{N1-1}\sum_{y=0}^{N2-1} I(x,y)\exp(-j\frac{2\pi ux}{N_1} - j\frac{2\pi vy}{N_2})$$

iar **transformata Fourier discreta inversa:**

$$I(x,y) = \frac{1}{N_1 * N_2}\sum_{u=0}^{N1-1}\sum_{v=0}^{N2-1} F(u,v)\exp(j\frac{2\pi ux}{N_1} + j\frac{2\pi vy}{N_2})$$

O prelucrare de imagine bazata pe transformata Fourier inseamna:

- Calculul transformatei F(u,v)
- Procesarea transformatei prin aplicarea unui filtru H(u,v) - H(u,v)*F(u,v)
- Calculul transformatei Fourier inverse a produsului H(u,v)*F(u,v)

Deci, *imaginea prelucrata* este:

$$I^{'}(x,y) = T^{-1}(\mathrm{H(u,v)} * \mathrm{F(u,v)})$$

unde F(u,v)=T(I(x,y))

Partea imaginara a functiei I' (x,y) se neglijeaza, fiind de regula foarte mica.

Definitie: Un *filtru trece jos,* H(u,v) este definit astfel:

H(u,v) = 1 daca $u^2 + v^2 < r^2$
H(u,v) = 0 daca $u^2 + v^2 \geq r^2$

ceea ce inseamna eliminarea frecventelor cu rata de ondulatie mai mare ca r din F(u,v) (care corespund la raze mai mari ca r):

daca $\sqrt{u^2 + v^2} > r$ atunci H(u,v)*F(u,v) = 0

Adesea, frecventele inalte reprezinta semnal fals in imagine (zgomot), indus de echipamentele electronice de producere a imaginilor discrete.

Zgomotul se reprezinta prin puncte negre in zone de imagine albe sau puncte albe in zone de imagine negre.

Definitie: Un *filtru trece sus,* H(u,v) este definit astfel:

H(u,v) = 0 daca $u^2 + v^2 < r^2$
H(u,v) = 1 daca $u^2 + v^2 \geq r^2$

Prin aplicarea unui filtru trece sus sunt eliminate frecvantele joase din F(u,v) (mai mici ca r):

daca $\sqrt{u^2 + v^2} < r$ atunci H(u,v)*F(u,v) = 0

Scopul aplicarii unui filtru trece sus este evidentierea detaliilor din imagine.

Pot fi folosite si **alte filtre**:

Pentru a rejecta o anumita banda de frecventa, $r_1 \leq r \leq r_2$

Pentru a retine doar o banda de frecvente

Pentru a retine sau elimina structuri coerente cu anumite orientari.

II. Tehnici de prelucrare a imaginilor color

Exista mai multe modele (sisteme) in care pot fi reprezentate culorile pixelilor: RGB, HSL sau HSV (Hue -nuanta, Saturation - saturatie, Value –intensitate), YIQ si altele.

Doua categorii de metode:

- Se proceseaza separat fiecare componenta a culorii si se compune rezultatul; in unele cazuri prelucrarea se aplica numai uneia dintre componente, de exemplu, intensitatea;
- Se lucreaza cu vectori de culoare (fiecare culoare are trei componente, fiind reprezentata printr-un vector in spatiul culorilor). De exemplu, culoarea unui pixel p(x,y) se reprezinta in spatiul (R,G,B) prin vectorul (Cr(x,y), Cg(x,y), Cb(x,y)).

1. Transformari de culoare

Fie o transformare de intensitate definita printr-o functie,

$g(x,y) = k^* f(x,y)$, $0< k < 1$, sau

$s = T(r)$, unde r este o culoare in imaginea de intrare iar s este culoarea in care este transformata r (in imaginea de iesire)

sau $(s1, s2, s3) = T(r1, r2, r3)$, unde ri si si sunt componentele celor 2 culori (ex. R,G,B).

Transformarea poate fi efectuata astfel:

- In HSV : $s3 = k^* r3$
- In RGB : $s1 = k^*r1$, $s2 = k^*r2$, $s3 = k^*r3$

Cu toate ca transformarea este mult mai simpla in HSV, conversia din RGB in HSV este mult mai costisitoare decat efectuarea transformarii in RGB.

Complementul unei imagini color

Fiecare culoare are o culoare complementara asa dupa cum un nivel de gri are un negativ (Neg(g) = Lmax – g). Aceasta proprietate este importanata in imbunatatirea detaliilor din zonele intunecate ale unei imagini color.

Culorile complementare culorilor R, G, B sunt, respectiv, C(Cyan), M(Magenta) si Y(Yellow).

- Calculul complementului unei imagini color inseamna inlocuirea fiecarei culori cu complementul sau.

2. Evidentierea culorilor de interes

- Este utila pentru evidentierea obiectelor din imagine.

Spatiul culorilor de interes dintr-o imagine poate fi reprezentat printr-un cub sau printr-o sfera.

De exemplu:

1. Cubul cu latura w si centrul in (a1,a2,a3)
2. Sfera de raza R si centrul in (a1,a2,a3)

(a1,a2,a3) poate fi media culorilor din subspatiul culorilor de interes.

Pixelii a caror culoare nu se incadreaza in volumul de interes sunt setati la o culoare "neutrala", pe care o notam cu n.

Atunci, transformarea si = T (ri) este:

1. si = n, daca (|rj – aj| > w/2), j=1 sau j=2 sau j=3
   ri, altfel
   (i=0,1, ….,Lmax)
2. si = n, daca $(r1-a1)^2 + (r2 - a2)^2 + (r3 - a3)^2 > R^2$
   ri, altfel
   (i=0,1, ….,Lmax)

3. Egalizarea histogramei

Poate fi efectuata ca si pentru imagini cu nivele de gri:

- Separat asupra fiecarei componente de culoare.
- Asupra componentei V (intensitate) din reprezentarea HSV a culorii, lasand culorile (componentele H si S) neschimbate.

4. Netezirea si marirea contrastului

Ca si in cazul imaginilor in nivele de gri, operatiile de netezire (smoothing) si marire a contrastului (sharpening) se implementeaza folosind filtre spatiale definite prin diferite masti de convolutie.

De exemplu, netezirea folosind filtrul medie aplicat fiecarei componente a culorii, pe o fereastra de 3x3 pixeli, este definita astfel :

$$g_\lambda(x,y) = 1/9 \sum_{i=-1}^{1} \sum_{j=-1}^{1} f_\lambda(x+i, y+j)$$