Lucrarea 6

Zgomotul în imagini

BREVIAR TEORETIC

Zgomotul este un semnal aleator, care afectează informaţia utilă conţinută într-o imagine. El poate apare de-alungul unui lanţ de transmisiune, sau prin codarea şi decodarea imaginii, şi reprezintă un element perturbator nedorit. De obicei se încearcă eliminarea lui prin diverse metode de filtrare.

Zgomotul se poate suprapune informaţiei utile în două moduri:

- **aditiv**. În acest caz, zgomotul se numeşte *zgomot aditiv* şi matematic se poate scrie:
$$g(x,y) = f(x,y) + n(x,y) \tag{6.1}$$
unde $f(x,y)$ este imaginea iniţială, neafectată de zgomot, $n(x,y)$ este zgomotul, iar $g(x,y)$ este imaginea afectată de zgomot.
- **multiplicativ**. În acest caz zgomotul se numeşte *zgomot multiplicativ*, iar fenomenul de suprapunere al acestuia peste informaţia utilă se scrie matematic astfel:
$$g(x,y) = f(x,y) \cdot n(x,y) \tag{6.2}$$
unde $f(x,y)$, $n(x,y)$ şi $g(x,y)$ au aceleaşi semnificaţii ca mai sus.

În continuare vom trata zgomotul ca fiind aditiv. Modelul de degradare a imaginii este reprezentat în figura 6.1. Zgomotul multiplicativ poate fi tratat la fel de simplu ca zgomotul aditiv, dacă logaritmăm relaţia (6.2):

$$log(g(x,y)) = log(f(x,y) \cdot n(x,y)) = log(f(x,y)) + log(n(x,y)) \tag{6.3}$$

Cantitativ, se poate aprecia măsura în care zgomotul a afectat informaţia utilă, calculând raportul semnal-zgomot[1]:

[1]SNR = (engl.) Signal Noise Ratio.

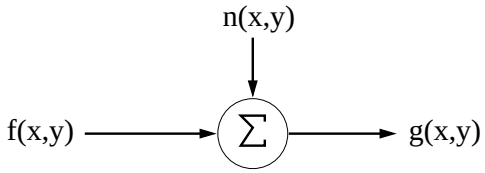

Figura 6.1: Modelul aditiv de degradare.

$$SNR = 10log\frac{\sum\limits_{i=0}^{M-1}\sum\limits_{j=0}^{N-1} f^2(i,j)}{\sum\limits_{i=0}^{M-1}\sum\limits_{j=0}^{N-1} n^2(i,j)}[dB] \tag{6.4}$$

$$SNR = 10log\frac{\sum\limits_{i=0}^{M-1}\sum\limits_{j=0}^{N-1} f^2(i,j)}{\sum\limits_{i=0}^{M-1}\sum\limits_{j=0}^{N-1} [f(i,j)-g(i,j)]^2}[dB] \tag{6.5}$$

Raportul semnal-zgomot reprezintă raportul dintre energia imaginii originale şi energia zgomotului suprapus acesteia. În continuare vom asocia valorilor pe care le ia zgomotul o variabilă aleatoare ξ, care, în funcţie de tipul zgomotului, va fi caracterizată de diferite funcţii de densitate de probabilitate.

6.1 Zgomotul cu distribuţie uniformă

Zgomotul cu distribuţie uniformă (vezi Figura 6.2) este caracterizat de o funcţie de densitate de probabilitate de forma:

$$w_\xi(x) = \begin{cases} A, & pentru\, x \in \left[-\frac{k}{2}; \frac{k}{2}\right], \\ 0, & în\, rest. \end{cases} \tag{6.6}$$

În Figura 6.3 puteţi observa efectele zgomotului cu distribuţie uniformă asupra imaginii “Lena”, pentru un raport semnal/zgomot de 5 dB.

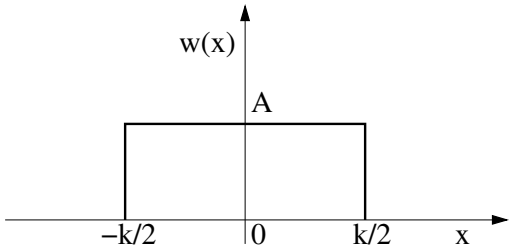

Figura 6.2: Funcţia de densitate de probabilitate pentru o distribuţie uniformă.

(a) (b)

Figura 6.3: Zgomotul uniform: (a) imaginea originală; (b) imaginea afectată de zgomot uniform, SNR=5 dB.

6.2 Zgomotul cu distribuţie gaussiană

Zgomotul gaussian este caracterizat de o funcţie de densitate de probabilitate de forma (vezi Figura 6.4):

$$w_\xi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{6.7}$$

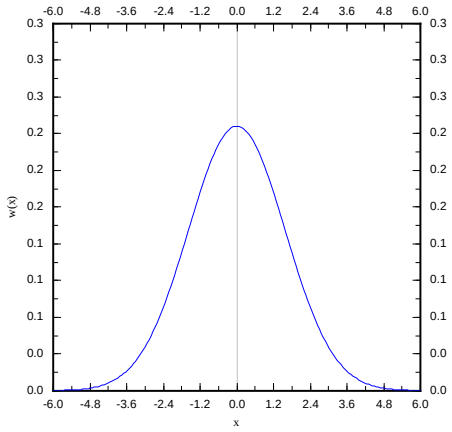

Figura 6.4: Funcţia de densitate de probabilitate pentru o distribuţie gaussiană.

Valoarea medie a unei variabilei aleatoare ξ, cu o distribuţie dată de funcţia $w_\xi(x)$ ca în relaţia (6.7), este μ, iar varianţa ei este σ^2. O distribuţie gaussiană se mai numeşte şi *normală*. O distribuţie normală de medie μ şi varianţă σ^2 se notează cu $N(\mu, \sigma^2)$.

În Figura 6.5 puteţi observa efectele zgomotului cu distribuţie gaussiană asupra imaginii "Lena", pentru un raport semnal/zgomot de 5 dB.

(a) (b)

Figura 6.5: Zgomotul Gaussian: (a) imaginea originală; (b) imaginea afectată de zgomot gaussian, SNR=5 dB.

6.3 Zgomotul de tip "sare şi piper"

După cum îi spune numele, acest tip de zgomot va afecta valorile pixelilor în două moduri: "sare" - adică noua valoare a pixelului va fi 255 (pixelul va fi alb), sau "piper" - adică noua valoare a pixelului va fi 0 (pixelul va fi negru). Zgomotul de tip "sare şi piper" (vezi Figura 6.6) este perfect decorelat, deoarece între valorile 0 şi 255, şi între coordonatele pixelilor afectaţi de zgomot nu există corelaţie.

(a) (b)

Figura 6.6: Zgomotul "sare şi piper": (a) imaginea originală; (b) imaginea cu 10% pixeli afectaţi de zgomot.

6.4 Alte tipuri de zgomot

Zgomotele diferă între ele în funcţie de distribuţia care le caracterizează. Alte funcţii de distribuţie utilizate sunt:

- distribuţia Rayleigh: $w_\xi(x) = xe^{-\frac{x^2}{2}}$
- distribuţia Maxwell: $w_\xi(x) = x^2e^{-\frac{x^2}{2}}$
- distribuţia Beta: $w_\xi(x) = x^b(1-x)^c$
- distribuţia Gamma (Erlang): $w_\xi(x) = x^ne^{-x}$
- distribuţia Laplace: $w_\xi(x) = e^{-|x|}$
- distribuţia Cauchy: $w_\xi(x) = \frac{1}{1+x^2}$

Aceste funcţii sunt reprezentate în Figura 6.7.

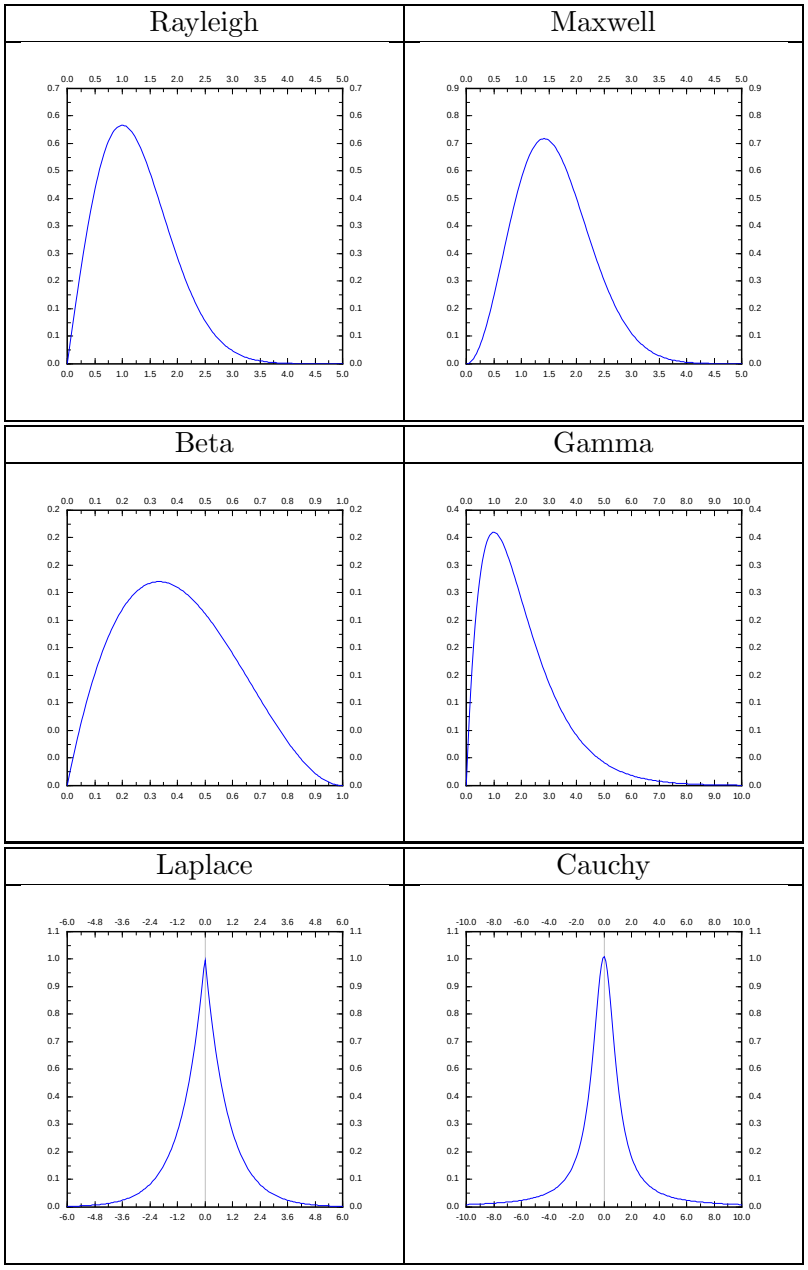

Figura 6.7: Diverse funcţii de distribuţie.

DESFĂŞURAREA LUCRĂRII

Problema 1. Suprapuneţi zgomot cu distribuţie uniformă peste o imagine în tonuri de gri (în meniul `Algoritmi`, funcţia `zgomot_uniform`).

```
void ImageViewer :: zgomot_uniform( void )
{
   int i, j;

   int w = image.width();
   int h = image.height();

   long int e_zgomot = 0; //energia zgomotului
   long int e_imagine = 0; //energia imaginii

   double SNR; //raportul semnal-zgomot

   //imaginea afectata de zgomot
   QImage image_aff( w, h, 32, 0, QImage::IgnoreEndian );

   int med = 0; //media zgomotului
   int dis = 200; //dispersia zgomotului

   for( i = 0; i < w; i++ )
      for( j = 0; j < h; j++ )
      {
         QRgb pixel = image.pixel( i, j );
         int nivel_gri = qRed( pixel );

         e_imagine += nivel_gri * nivel_gri;

         //srand( rand() );
         int zgomot = ( int )( med + sqrt( 3 )*
            (2. * ( rand()/( RAND_MAX + 1. )))*sqrt(dis) );
         e_zgomot += zgomot * zgomot;
         int val = nivel_gri + zgomot;

         if( val > 255 )
            val = 255;
         if( val < 0 )
            val = 0;

       image_aff.setPixel( i, j, qRgb( val, val, val ));
   }

   SNR = 10 * log( 1. * e_imagine / e_zgomot );

   image = image_aff;
   pm = image;
```

```
   update();

   QString mesaj;
   mesaj.sprintf( "SNR = %6.3lf dB", SNR );
   QMessageBox::about( this, "SNR", mesaj );
}
```

Problema 2. Modificaţi media şi dispersia zgomotului cu distribuţie uniformă (în fişierul `algoritmi.cpp`, funcţia `generează_zgomot_uniform`). Observaţi efectele.

Problema 3. Suprapuneţi zgomot cu distribuţie gaussiană peste o imagine în tonuri de gri (în meniul `Algoritmi`, funcţia `zgomot_gaussian`). Diferenţa între această funcţie şi cea care generează constă în modul în care este calculată valoarea zgomotului, şi anume:

```
int zgomot = ( int )( med + sqrt( 2 ) *
   sqrt( -1.* log( rand()/( RAND_MAX + 1. ))) *
   cos( 2 * 3.1415926 * ( rand()/( RAND_MAX+1 ))) *
   sqrt( dis ) );
```

Problema 4. Modificaţi media şi dispersia zgomotului cu distribuţie gaussiană (fişierul `algoritmi.cpp`, funcţia `generează_zgomot_gaussian`). Observaţi efectele.

Problema 5. Suprapuneţi zgomot de tip "sare şi piper" peste o imagine în tonuri de gri (în meniul `Algoritmi`, funcţia `zgomot_salt_and_pepper`). Codul acesteia este prezentat în continuare:

```
void ImageViewer :: salt_and_pepper( void )
{
   int i, j, k;
   int w = image.width();
   int h = image.height();

   double nr = 0.1; //procentul de pixeli afectati de zgomot

   srand( rand() );

   k = 0;
   while( k < ( int )( w * h * nr ) )
   {
      i = ( int )( 1. * w * rand() / ( RAND_MAX + 1. ) );
      j = ( int )( 1. * h * rand() / ( RAND_MAX + 1. ) );

      QRgb sare_piper;
```

```
      if( ( 100. * rand() / ( RAND_MAX + 1. ) ) >= 50 )
         sare_piper = qRgb( 255, 255, 255 );
      else
         sare_piper = qRgb( 0, 0, 0 );

      if( (i >= 0) && (i < w) && (j >= 0) && (j < h) )
         image.setPixel( i, j, sare_piper );

      k++;
   }

   pm = image;
   update();
}
```

Problema 6. Modificaţi procentul de pixeli afectaţi de zgomot de tip "sare şi piper" (fişierul `algoritmi.cpp`, funcţia `zgomot_salt_and_pepper`). Observaţi efectele.