

Lucrarea 3

Îmbunătățirea imaginilor prin operații punctuale

BREVIAR TEORETIC

Termenul general de îmbunătățire a imaginilor se referă la o clasă largă de operații, ce au ca scop mărirea detectabilității componentelor imaginii. Această detectabilitate depinde în primul rând de percepția vizuală a unui observator uman și deci reprezintă o apreciere subiectivă a imaginii.

Îmbunătățirea calității unei imagini se face fără a presupune vreun model de degradare sau a lua în considerare vreo informație legată de imaginea originală. Paradoxal, dar și o imagine originală (nedegradată) poate fi îmbunătățită.

În general, îmbunătățirea se referă la accentuarea unor caracteristici ale imaginii, cum ar fi muchiile, contururile sau contrastul. Procesul de îmbunătățire nu mărește cantitatea de informație conținută de o imagine.

3.1 Operațiile punctuale

Operațiile punctuale sunt definite de o funcție, care atribuie un nou nivel de gri pixelilor din imagine. Noua valoare a pixelului va depinde doar de vechea valoare a acestuia, de unde și denumirea de “operație punctuală”. Matematic se poate scrie:

$$g(k, l) = \phi(f(k, l)) \quad (3.1)$$

unde f este imaginea originală, iar g imaginea îmbunătățită. $g(k, l)$ reprezintă noua valoare a pixelului (k, l) , iar $f(k, l)$ vechea valoare. Operația punctuală este descrisă de funcția ϕ .

O operație sau o transformare punctuală este reprezentată în Figura 3.1.

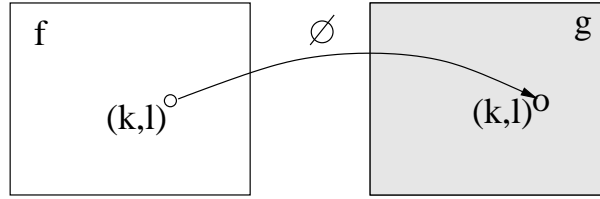


Figura 3.1: Reprezentarea unei operații punctuale.

3.2 Accentuarea contrastului

Această operație se folosește în special pentru îmbunătățirea imaginilor cu contrast scăzut. Acesta poate apărea datorită unei slabe iluminări, a unei iluminări neuniforme sau din cauza unor neliniarități ale caracteristicilor unui senzor de captură a imaginii. O funcție tipică de accentuare a contrastului are următoarea formă matematică:

$$\phi(x) = \begin{cases} \alpha x, & \text{pentru } x \in [0, a) \\ \beta(x - a) + v_a, & \text{pentru } x \in [a, b) \\ \gamma(x - b) + v_b, & \text{pentru } x \in [b, L) \end{cases} \quad (3.2)$$

unde L reprezintă numărul de nivele pe care se face cuantizarea tonurilor de gri. Cazul cel mai frecvent este $L = 256$, pentru o cuantizare pe 8 biți, x luând valori în intervalul $[0, 255]$. Valorile α, β și γ reprezintă pantele celor trei segmente de dreaptă.



Figura 3.2: Funcția de accentuare de contrast.

După cum se observă din Figura 3.2, pantele α și γ sunt pozitive și subunitare, iar panta β este pozitivă și supraunitară. Un segment cu pantă subunitară realizează o “apropiere” a nivelelor de gri, pe când un segment

cu pantă supraunitară va realiza o “depărtare” a nivelelor de gri, și deci o accentuare a contrastului.

3.3 Întinderea maximă a contrastului

Aceasta este un caz particular al operației de accentuare. Se folosește, în general, pentru reducerea zgomotului dintr-o imagine, atunci când se știe că acesta ia valori cu precădere în intervalul $[a, b]$. Forma matematică este următoarea:

$$\phi(x) = \begin{cases} 0, & \text{pentru } x \in [0, a) \\ \beta(x - a), & \text{pentru } x \in [a, b] \\ L - 1, & \text{pentru } x \in (b, L) \end{cases} \quad (3.3)$$

Nivelele de gri aflate în intervalul $[a, b]$ vor fi distanțate, ca urmare a pantei supraunitare, iar nivelele de gri ce se găsesc în afara acestui interval, vor fi înlocuite fie cu alb, fie cu negru, după caz.

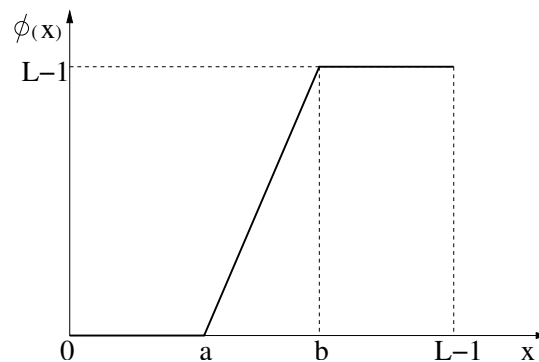


Figura 3.3: Funcția de întindere maximă a contrastului.



Figura 3.4: Întinderea maximă a contrastului: (a) imaginea originală și (b) imaginea rezultată, pentru $a=50$, $b=200$.

3.4 Binarizarea imaginilor

Binarizarea sau prăguirea (“thresholding”) imaginilor este un caz special al întinderii maxime a contrastului, pentru care $a = b$. Rezultatul operației de binarizare este o imagine care conține doar două nivele de gri: alb și negru.

Pentru imagini în tonuri de gri, operația de binarizare se scrie matematic astfel:

$$\phi(x) = \begin{cases} 0, & \text{pentru } x < T \\ L - 1, & \text{pentru } x \geq T \end{cases} \quad (3.4)$$

unde T este o valoare de prag, reprezentând o valoare întreagă din intervalul $[0, L)$.

Pentru imaginile color, binarizarea se poate face în următorul mod:

$$\phi(v) = \begin{cases} 0, & \text{pentru } Y(v) < T \\ L - 1, & \text{pentru } Y(v) \geq T \end{cases} \quad (3.5)$$

unde v este un vector tridimensional ce reprezintă culoarea pixelului (de exemplu $v = (R, G, B)$), iar $Y(v)$ reprezintă luminanța ($Y = 0.3R + 0.6G + 0.1B$).

În urma acestei transformări, contrastul este maximizat la nivelul întregii imagini.

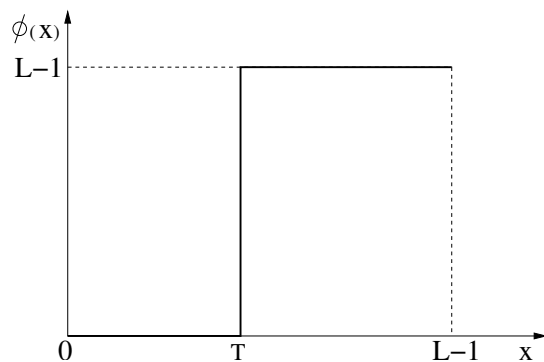


Figura 3.5: Funcția de binarizare.

Figura 3.6: Binarizarea: (a) imaginea originală și (b) imaginea binarizată cu $T=127$.

3.5 Negativarea imaginilor

Negativul unei imagini se obține prin inversarea ordinii nivelelor de gri. Pentru imaginile în tonuri de gri, operația de negativare se face folosind funcția:

$$\phi(x) = (L - 1) - x \quad (3.6)$$

reprezentată în Figura 3.7, iar pentru imaginile color:

$$\phi(v) = ((L - 1) - R, (L - 1) - G, (L - 1) - B) \quad (3.7)$$

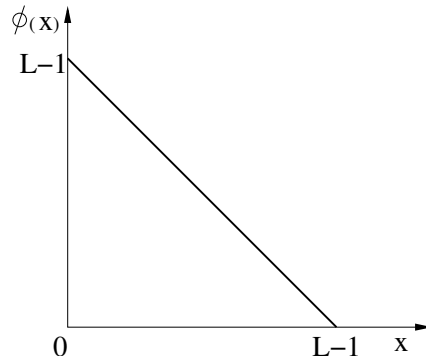


Figura 3.7: Funcția de negativare.

Negativarea imaginilor este utilă pentru afișarea unor imagini medicale sau pentru realizarea de imagini negative pe suporturi fizice, de exemplu, de tip peliculă.



Figura 3.8: Negativarea: (a) imaginea originală și (b) negativul imaginii.

3.6 Decuparea imaginilor

Decuparea cu păstrarea fundalului este dată de formula:

$$\phi(x) = \begin{cases} L - 1, & \text{pentru } x \in [a, b] \\ x, & \text{în rest.} \end{cases} \quad (3.8)$$

iar decuparea fără păstrarea fundalului este dată de formula:

$$\phi(x) = \begin{cases} L-1, & \text{pentru } x \in [a, b] \\ 0, & \text{în rest.} \end{cases} \quad (3.9)$$

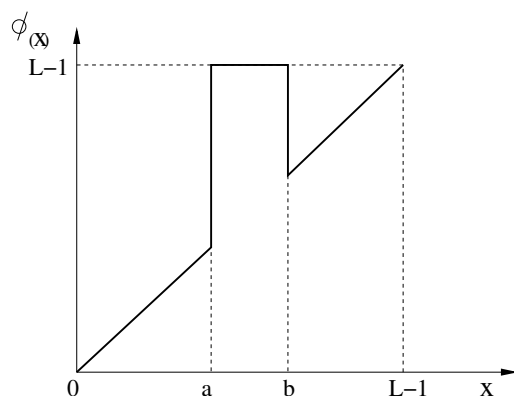


Figura 3.9: Funcția de decupare cu păstrarea fundalului.

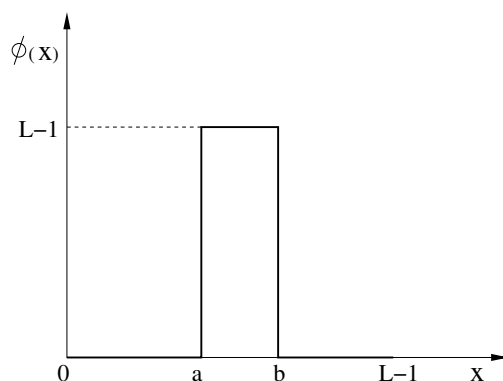


Figura 3.10: Funcția de decupare fără păstrarea fundalului.

Aceste operații permit “decuparea” unor regiuni din imagine, caracterizate de anumite nivele de gri. Acest lucru este folosit atunci când diferite caracteristici ale imaginii sînt conținute în nivelele de gri respective, cum ar fi de exemplu decuparea regiunilor de temperatură joasă reprezentate de nori din imaginile obținute de un satelit meteo. În astfel de imagini, nivelele de gri ce corespund unor nori sînt direct proporționale cu valorile de temperaturi joase.

DESFĂȘURAREA LUCRĂRII

Problema 1. Compilați sursele C++ ale lucrării. Rulați aplicația și

observați rezultatul accentuării contrastului pentru o imagine în tonuri de gri (lena_AN.bmp). Funcția care realizează accentuarea contrastului este prezentată în continuare, pentru următoarele valori: $a = 80$, $b = 170$, $V_a = 20$ și $V_b = 235$.

Pantele care caracterizează funcția de accentuare a contrastului vor fi următoarele: $\alpha = \frac{20}{80} = 0.4$, $\beta = \frac{235-20}{170-80} = \frac{215}{90}$ și $\gamma = \frac{255-235}{255-170} = \frac{20}{85}$.

```
int ImageViewer :: f_accentuare( int nivel_gri )
{
    if( nivel_gri >= 0 && nivel_gri <= 80 )
        return ( int )( 0.4 * nivel_gri );

    if( nivel_gri > 80 && nivel_gri <= 170 )
        return ( int )( 215 / 90. * ( nivel_gri - 80 ) + 20 );

    if( nivel_gri > 170 && nivel_gri <= 255 )
        return ( int )( 20 / 85. * ( nivel_gri - 170 ) + 235 );

    return nivel_gri;
}

void ImageViewer :: accentueaza_contrastul( void )
{
    int w, h;
    int i, j;

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

    QImage imag_acc( w, h, 32, 0, QImage :: IgnoreEndian );

    for( i = 0; i < w; i++ )
        for( j = 0; j < h; j++ )
        {
            QRgb pixel = image.pixel( i, j );
            int gri_vechi = qRed( pixel );
            int gri_nou = f_accentuare( gri_vechi );
            imag_acc.setPixel( i, j,
                qRgb( gri_nou, gri_nou, gri_nou ) );
        }

    QImageIO iio;
    iio.setImage( imag_acc );
}
```



```

    iio.setFileName( "imag_acc.bmp" );
    iio.setFormat( "BMP" );
    iio.write();
}

```

Problema 2. Modificați valorile a, b, V_a și V_b ale funcției de accentuare a contrastului, și observați rezultatele accentuării contrastului.

Problema 3. Implementați operația de întindere maximă a contrastului.

Problema 4. Implementați operația de binarizare. Observați rezultatele acesteia pentru diferite valori ale pragului T .

Problema 5. Observați rezultatul operației de negativare, pentru o imagine în tonuri de gri și pentru o imagine color.

```

void ImageViewer :: negativeaza_imaginea( void )
{
    int w, h;
    int i, j;

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

    QImage imag_neg( w, h, 32, 0, QImage :: IgnoreEndian );

    for( i = 0; i < w; i++ )
        for( j = 0; j < h; j++ )
        {
            QRgb pixel = image.pixel( i, j );
            int r = qRed( pixel );
            int g = qGreen( pixel );
            int b = qBlue( pixel );
            imag_neg.setPixel( i, j, qRgb(255-r, 255-g, 255-b) );
        }

    QImageIO iio;
    iio.setImage( imag_neg );
    iio.setFileName( "imag_neg.bmp" );
    iio.setFormat( "BMP" );
    iio.write();
}

```

Problema 6. Implementați cele două operații de decupare.

