

Ψηφιακή Επεξεργασία Εικόνας
Εργασία 3: Πολυδιακριτική Ανάλυση-Συστοιχίες
Φίλτρων Τέλειας Ανακατασκευής

Ζαμπόκας Γεώργιος 7173

Έχουμε υλοποιήσει τις συναρτήσεις:

```
[ y0,y1 ] = analysis1d( h0,h1,x )
```

Η συνάρτηση ανάλυσης μονοδιάστατου σήματος. Παίρνει ως όρισμα ένα σήμα-γραμμή και υποδειγματολειτουργεί 1:2 την συνέλιξή του με τα φίλτρα αποσύνθεσης h_0, h_1 . Τα δύο μονοδιάστατα σήματα μισοού μήκους που προκύπτουν αποθηκεύονται στα y_0, y_1 αντίστοιχα.

```
[ xhat ] = synthesis1d( g0,g1,y0,y1 )
```

Η συνάρτηση σύνθεσης δύο μονοδιάστατων σημάτων σε ένα με διπλάσιο μήκος. Αρχικά, υπερδειγματολειτουργεί τα δοσμένα σήματα 2:1 εκτελεί την συνέλιξή τους με τα φίλτρα ανακατασκευής g_0, g_1 . Τα αποτελέσματα προστίθενται και επιστρέφονται στην μεταβλητή $xhat$.

Στο `demo3.m` φαίνεται η επίδειξη της επεξεργασίας της εικόνας:

```
x=imread('farm.jpg');
```

Διαβάζουμε την δοσμένη εικόνα.

```
load('db10.mat');  
h=db10;
```

Διαβάζουμε τον δοσμένο πίνακα και τον περνάμε σε μεταβλητή h . Ο πίνακας αυτός αποτελεί και το h_0 (high decomposition filter)

```
[ h0,h1,g0,g1 ] = orthonormalFilterBanks( h );
```

Η συνάρτηση αυτή δέχεται ως όρισμα το h δηλαδή το h_0 και μέσω των σχέσεων παράγει τα αντίστοιχα h_1, g_0, g_1 . Οι σχέσεις υλοποιούνται στο πεδίο του χρόνου.

```
[ y00,y01,y10,y11 ] = analysis2d( h0,h1,x );
```

Η συνάρτηση αυτή εκτελεί την αποσύνθεση της εικόνας σε 4 εικόνες με διαστάσεις $M/2 \times N/2$ (οταν οι διαστάσεις της αρχικής είναι $M \times N$).

Αρχικά, καλεί την `analysis1d` N -φορές για ανάλυση και έτσι προκύπτουν 2 εικόνες διάστασης $M/2 \times N$.

Έπειτα, με $(M/2)$ -κλήσεις της `analysis1d` για την καθεμία από τις παραπάνω εικόνες προκύπτουν τελικά 4 εικόνες διάστασης $M/2 \times N/2$ (`y00,y01,y10,y11`).

Οι 4 αυτές εικόνες επιστρέφονται από την συνάρτηση.

(Όλα τα σήματα που δίνονται ως ορίσματα στην `analysis1d` είναι πίνακες-γραμμές καθώς έτσι έχει φτιαχτεί ο κώδικας).

```
[ xhat ] = synthesis2d( g0,g1,y00,y01,y10,y11 );
```

Η συνάρτηση σύνθεσης των τεσσάρων εικόνων $M/2 \times N/2$ σε μία, ίδια με την αρχική εικόνα x .

Ξεκινώντας από τις 4 εικόνες καλούμε ανά ζεύγη την `synthesis1d` $M/2$ -φορές για τις `y00,y01` και `y10,y11`. Έτσι, προκύπτουν 2 εικόνες διάστασης $M/2 \times N$.

Για αυτές τις δύο εικόνες καλούμε την `synthesis1d` N -φορές και προκύπτει μία εικόνα διάστασης της αρχικής $M/2 \times N/2$.

(Όλα τα σήματα που δίνονται ως ορίσματα στην `synthesis1d` είναι πίνακες-γραμμές καθώς έτσι έχει φτιαχτεί ο κώδικας).

Τέλος, βρίσκουμε τη μέγιστη τιμή φωτεινότητας L στην ανακατασκευασμένη εικόνα και πραγματοποιούμε κανονικοποίηση σε αυτή, διαιρώντας όλα τα στοιχεία της με $L-20$, καθώς το αρχικό αποτέλεσμα είναι υπερβολικά φωτεινό.

```
[m1,n1]=size(x);
```

```
max=x(1,1);
```

```
for i=1:m1
```

```
    for j=1:n1
```

```
        if x(i,j)>max
```

```
            max=x(i,j);
```

```
        end
```

```
    end
```

```
end
```

```
L=double(max);
```

Η παραπάνω διπλή επανάληψη εντοπίζει με διαδοχικές συγκρίσεις (όπως έγινε και στην

synthesis2d) τη μέγιστη τιμή της φωτεινότητας L στην αρχική εικόνα.

```
figure(1)
subplot(221)
imshow(y00/L);      (*)
subplot(222)
imshow(y01);        (*)
subplot(223)
imshow(y10);        (*)
subplot(224)
imshow(y11);        (*)
figure(2)
subplot(211)
imshow(x);
subplot(212)
imshow(xhat);
```

(*) Έχει παραλειφθεί η κανονικοποίηση των αποτελεσμάτων y_{01}, y_{10}, y_{11} επειδή με κανονικοποίηση φωτεινότητας εμφανίζονται ως τελείως μαύρες εικόνες αντίθετα με την y_{00} η οποία είναι ιδιαίτερα φωτεινή χωρίς την κανονικοποίηση.

Τα αποτελέσματα της επεξεργασίας παρουσιάζονται παρακάτω:

γ00/L:



y01:



y10:



y11:



Η αρχική x:



Η ανακατασκευασμένη xhat:

