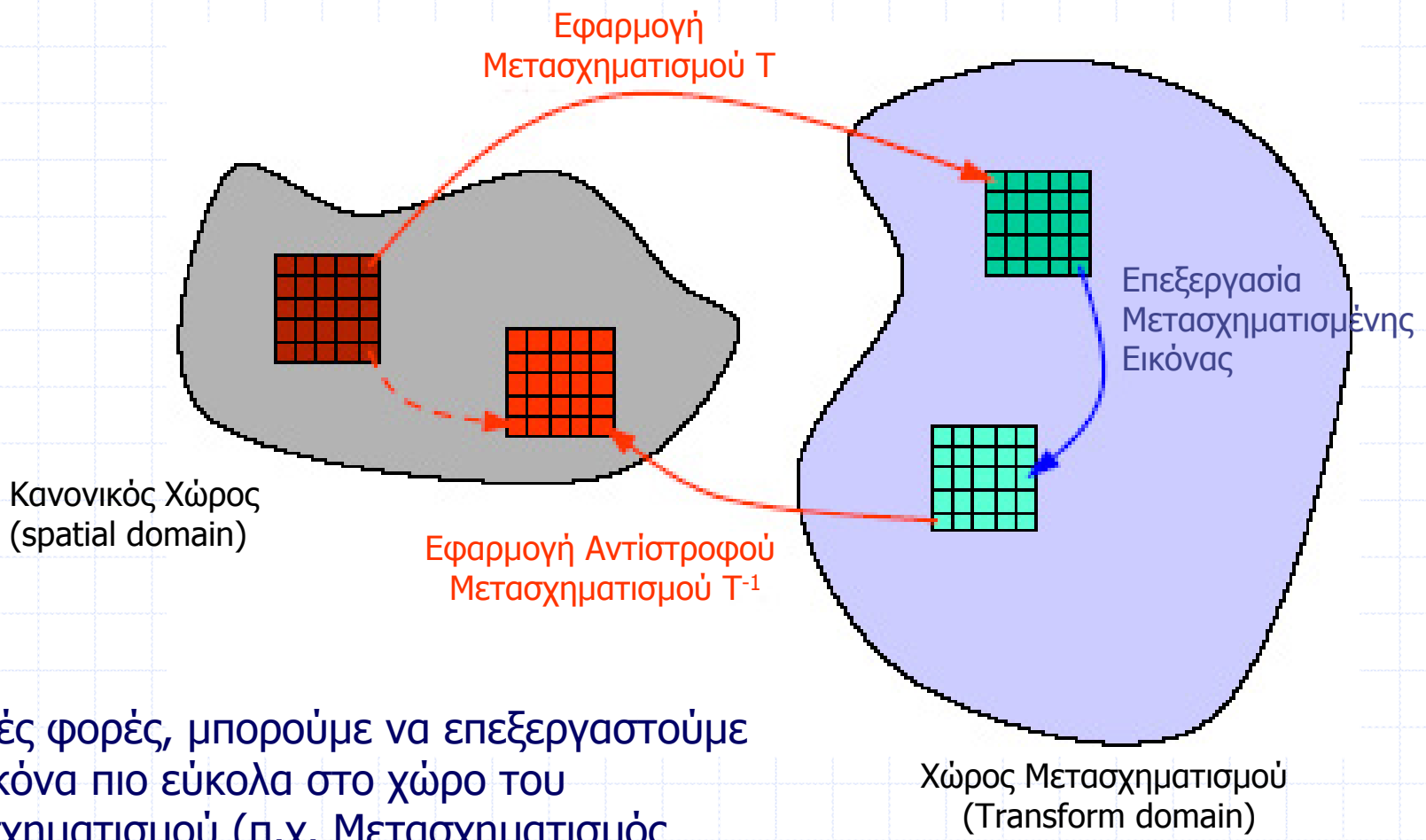




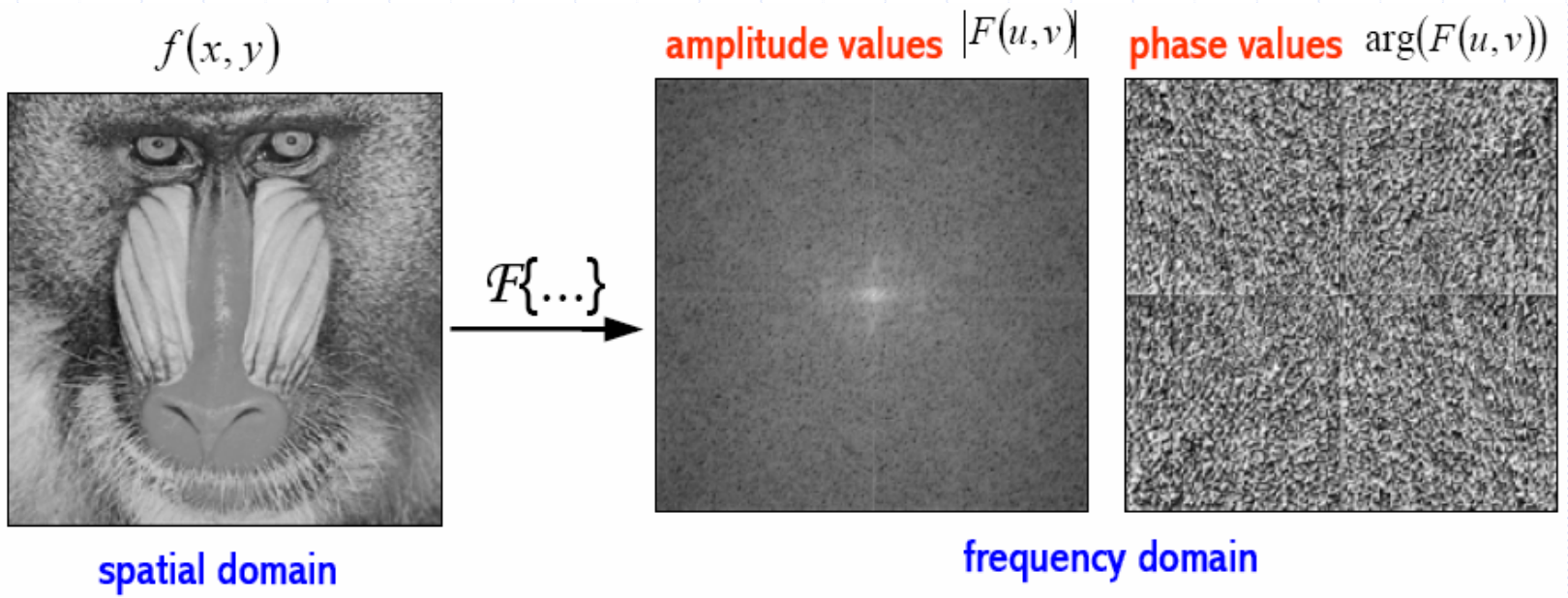
# ΣΥΝΟΛΙΚΟΙ ΤΕΛΕΣΤΕΣ



Μερικές φορές, μπορούμε να επεξεργαστούμε μια εικόνα πιο εύκολα στο χώρο του μετασχηματισμού (π.χ. Μετασχηματισμός Fourier: χώρος συχνοτήτων)



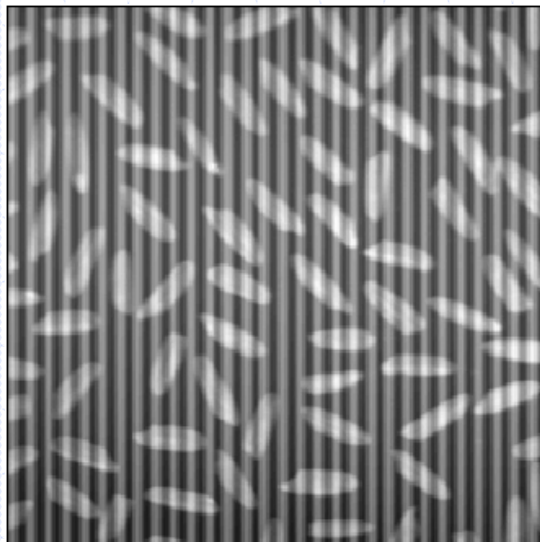
# Φάσμα Συχνοτήτων Εικόνας (II)



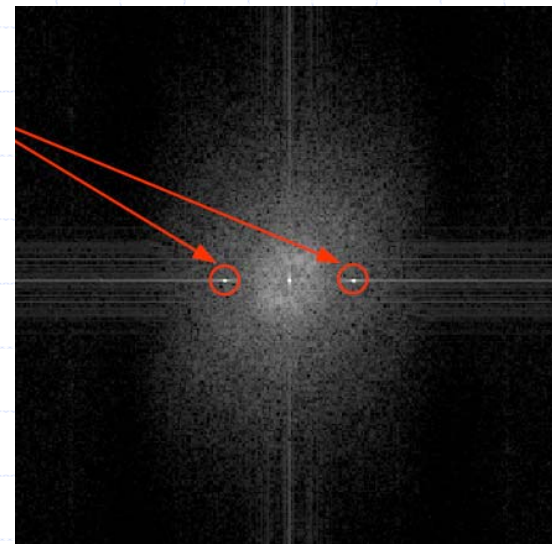


# Απομάκρυνση Οργανωμένου Θορύβου (I)

- ◆ Σε κάποιες περιπτώσεις η εικόνα περιέχει περιοδικό θόρυβο. Εδώ έχουμε θόρυβο συνημίτονου. Εφαρμόζουμε συνολικό τελεστή
- ◆ Μετασχηματίζοντας την εικόνα στον χώρο των συχνοτήτων μπορούμε να εντοπίσουμε τις εντάσεις της ενέργειας που προκαλούν την παραμόρφωση



FFT  
→





# Ορισμός τοπικού τελεστή

window

2	4	6	7	3	0	3
4	3	2	9	4	5	8
3	2	0	0	1	4	5
6	7	8	5	2	3	4
5	6	7	2	1	2	0
4	4	5	1	0	5	1
2	6	6	0	2	6	7

Source image

Το παράθυρο συνέλιξης κινείται σε όλη την περιοχή της εικόνας

Filter Kernel

0	-1	0
-1	4	-1
0	-1	0

⊗

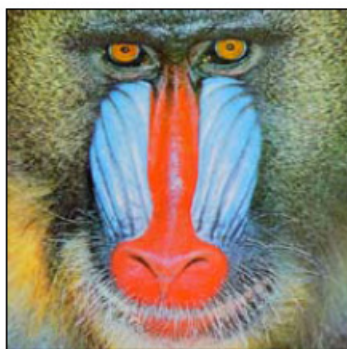
=

Pixel „5“ becomes „8“

8

$$(0 \cdot (-1) + 8 \cdot (-1) + 5 \cdot 4 + 2 \cdot (-1) + 2 \cdot (-1)) = 8$$

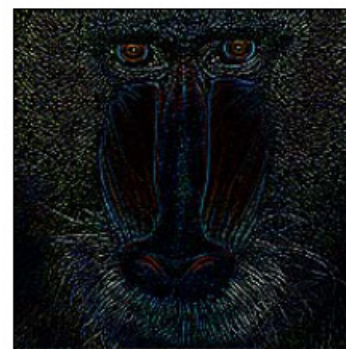
2-dimensional convolution operator



⊗

0	-1	0
-1	4	-1
0	-1	0

=

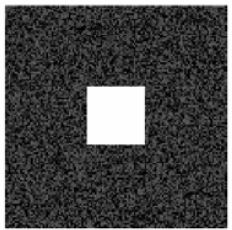




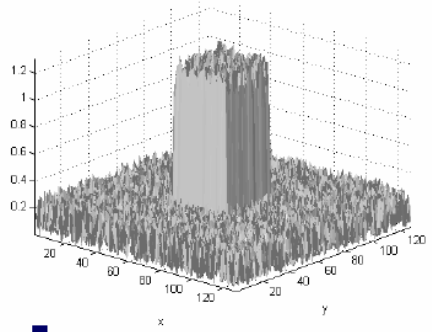


# Gaussian Filter (II)

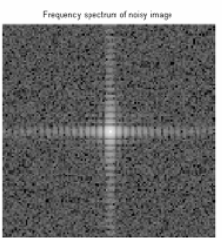
high frequencies are damped uniformly



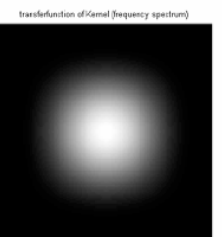
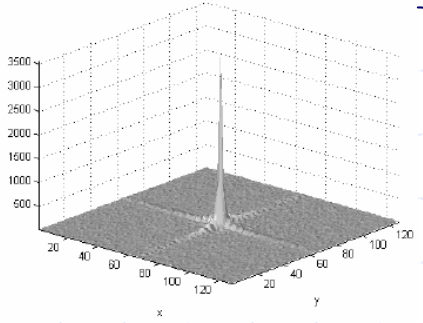
Noisy Image in spatial domain



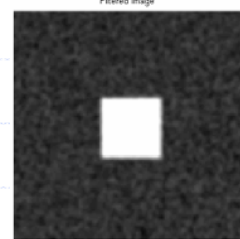
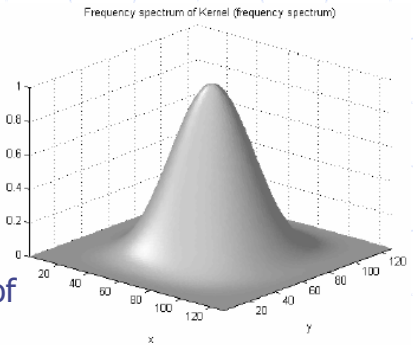
FFT



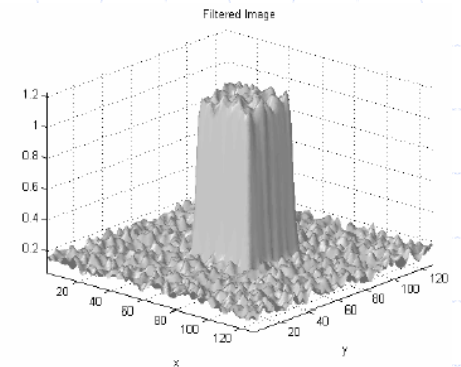
Frequency spectrum of noisy image



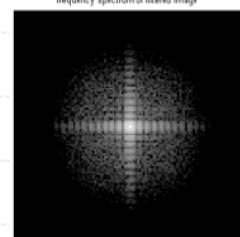
Frequency spectrum of 5x5 Gaussian Filter



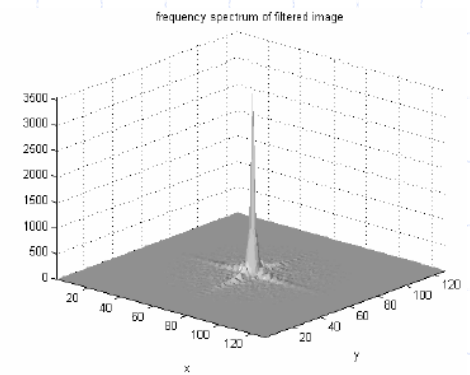
Filtered image in spatial domain



IFFT



Frequency spectrum of filtered image:







# Συμπίεση Εικόνας

# Επισκόπηση Συμπίεσης Εικόνας

## ◆ Χωρίς Απώλεια Πληροφορίας (μη απωλεστικοί)

- Η εικόνα αποσυμπιέζεται χωρίς απώλεια πληροφορίας

## ◆ Εφαρμογή:

- Όταν η δημιουργία της εικόνας είναι δύσκολη (χρόνος, διαδικασία)
- Όταν σημαντική πληροφορία μπορεί να χαθεί χρησιμοποιώντας αλγόριθμους απώλειας πληροφορίας (ιατρικές εικόνες)

## ◆ Με Απώλεια Πληροφορίας (απωλεστικοί)

- Η εικόνα αποσυμπιέζεται με απώλεια πληροφορίας
- Γενικά επιτυγχάνεται μεγαλύτερος λόγος συμπίεσης

## ◆ Εφαρμογή:

- Όταν η δημιουργία της εικόνας είναι εύκολη
- Όταν (μεγάλες) εικόνες μεταφέρονται από τηλεπικοινωνιακά κανάλια (video conferencing)





## Run-length Encoding (II)

- ◆ Άσκηση: Πόσα bits χρειάζονται για την κωδικοποίηση της παρακάτω 1 bit εικόνας χρησιμοποιώντας Κωδικοποίηση Τρέχοντος Μήκους;

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

requires  $8 \cdot 12 = 96$  bits  
uncompressed

**Solution:**

Run-length encoded:

5 2 10 2 10 2 6 10 2

10 6 2 10 2 10 2 5

→  $17 \cdot 4$  bit = 68 bit

→ 30% reduction



# Huffman Coding (I)

- ◆ Διαδική κωδικοποίηση συνήθως θέτει ίσου μεγέθους κώδικο-λέξεις ανεξάρτητα από την πιθανότητα εμφάνισης τους.
- ◆ **Huffman Coding** : Λέξεις με μεγαλύτερη πιθανότητα εμφάνισης έχουν λιγότερο αριθμό bits

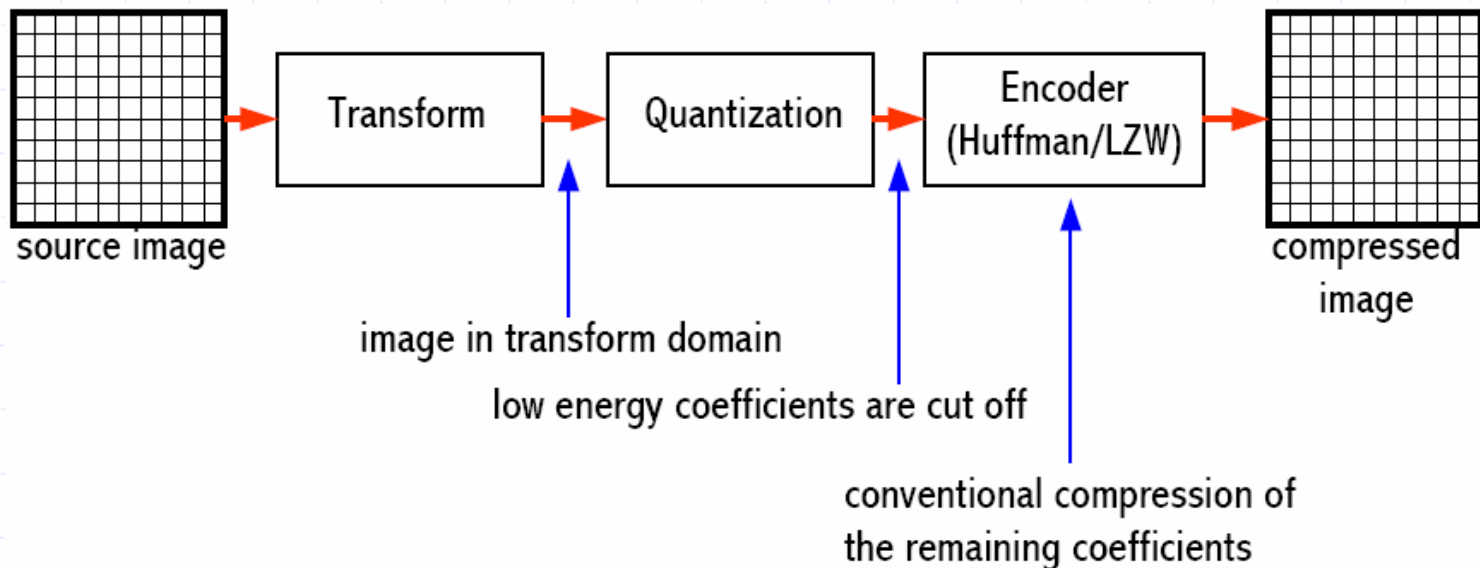
**Example:** she sells sea shells at the sea shore.

	s	e	'	l	h	a	t	o	r
#	8	7	7	4	4	3	2	1	1
$p_i$	0.21	0.19	0.19	0.11	0.11	0.08	0.05	0.03	0.03
Code:	00	01	110	100	101	1110	11110	111110	111111

most frequent  
→ shortest code word

# Κωδικοποίηση Εικόνας με Μετασχηματισμό

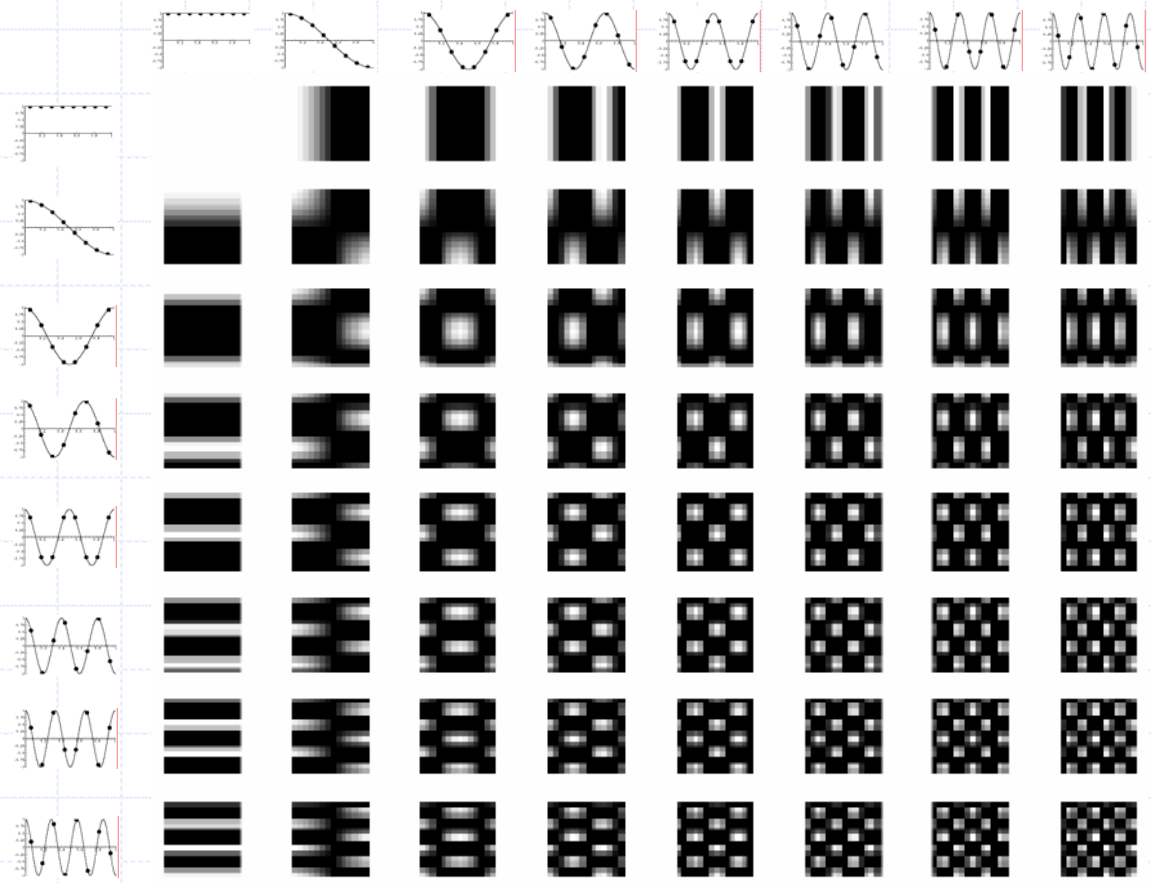
- ◆ Επιλέγουμε κατάλληλο μετασχηματισμό ώστε οι συναρτήσεις βάσης να συγκεντρώνουν όσο το δυνατό περισσότερη πληροφορία σε λιγότερους συντελεστές -> Συγκέντρωση ενέργειας -> συντελεστές με χαμηλή ενέργεια απορρίπτονται -> Συμπίεση
- ◆ Κβαντοποίηση και κωδικοποίηση των συντελεστών





# Discrete Cosine Transform

◆ 64 Συναρτήσεις βάσης DCT ενός 8x8 block:





## JPEG Συμπίεση (III)

- ◆ Οι AC συντελεστές (οι περισσότεροι έχουν τιμή μηδέν) κωδικοποιούνται με Run-length Coding
- ◆ Οι DC (difference coefficients) συντελεστές κωδικοποιούνται με Huffman Coding.

Processing of 8x8 block image:

