



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**Πρόγραμμα Μεταπτυχιακών Σπουδών**  
**Επιστήμη και Τεχνολογία της Πληροφορικής και**  
**των Υπολογιστών**  
**Ειδίκευση Λογισμικού και Πληροφοριακών Συστημάτων,**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Τεχνικές αραιής κωδικοποίησης και οι εφαρμογές τους στην**  
**αποθρομβοποίηση και αποκατάσταση ψηφιακών εικόνων**

**Παναγιώτα Γ. Ρούνη**  
**A.M. mcse\_19001**

**Επιβλέπων: Αναστάσιος Κεσίδης, Αναπληρωτής Καθηγητής**



## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Τεχνικές αραιής κωδικοποίησης και οι εφαρμογές τους στην  
αποθρομβοποίηση και αποκατάσταση ψηφιακών εικόνων**

**Παναγιώτα Γ. Ρούνη**

**A.M. mcse\_19001**

**Επιβλέπων:**

**Αναστάσιος Κεσίδης, Αν. Καθηγητής**

**Εξεταστική Επιτροπή:**

**Πάρις Μαστοροκώστας, Καθηγητής**

**Αθανάσιος Βουλόδημος, Επ. Καθηγητής**

**Ημερομηνία εξέτασης: 6/5/2022**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη

Παναγιώτα Ρούνη του Γεωργίου, με αριθμό μητρώου mcse\_19001 φοιτήτρια του Προγράμματος Μεταπτυχιακών Σπουδών Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι ..... και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Η Δηλούσα





## Ευχαριστίες

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της επεξεργασίας εικόνας και αναγνώρισης προτύπων. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου κ. Κεσίδης Αναστάσιος, τον οποίο θα ήθελα να ευχαριστήσω.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για τη στήριξη της σε όλη αυτή την πορεία των ακαδημαϊκών μου χρόνων.





## Περίληψη

Η αραιή κωδικοποίηση (sparse coding) έχει αποδειχθεί ένα εξαιρετικά ισχυρό εργαλείο για την ανάκτηση, αναπαράσταση αλλά και συμπίεση πολυδιάστατων δεδομένων. Τα σήματα αντιπροσωπεύονται από γραμμικούς συνδυασμούς μερικών στοιχείων ενός λεξικού που δεν είναι σταθερό και προκαθορισμένο, αλλά προκύπτει μέσω διαδικασίας μάθησης από δεδομένα εκπαίδευσης. Οι παραγόμενες συμπαγείς αναπαραστάσεις των δεδομένων μπορούν να εξοικονομήσουν μνήμη και υπολογιστικό χρόνο και παράλληλα να βελτιώσουν την απόδοση των αλγορίθμων που εφαρμόζονται στα δεδομένα αυτά. Η παρούσας διπλωματική εργασία έχει στόχο την μελέτη των τεχνικών αραιής κωδικοποίησης σε εφαρμογές επεξεργασίας εικόνας, όπως είναι η αποθρομβοποίηση και η ανάκτηση εικόνας. Μελετήθηκε η χρήση του αλγορίθμου k-SVD για την επεξεργασία έγχρωμων και μη εικόνων με στόχο την ελαχιστοποίηση ή αφαίρεση θορύβου από αυτές καθώς επίσης και η ανάκτηση και αποκατάσταση σε χαμένη ή κατεστραμμένη πληροφορία της εικόνας με την χρήση του ίδιου αλγορίθμου.

## Abstract

Sparse coding has proven to be an extremely powerful tool for retrieving, representing and compressing multidimensional data. The signals are represented by linear combinations of some elements of a dictionary that is not fixed and predefined, but is obtained through a learning process based on training data. The resulting solid representations of the data can save memory and computing time and also improve the performance of the algorithms applied. The present dissertation aims to study sparse coding techniques in image processing applications, such as noise reduction and image restoration. The use of the k-SVD algorithm for the processing of color and non-color images in order to minimize or remove noise from digital images as well as the recovery and restoration of lost or damaged image information using the same algorithm was studied.

## Περιεχόμενα

|  |    |
|--|----|
| Ευχαριστίες.....   | 7  |
| Περίληψη.....  | 9  |
| Abstract .....   | 10 |
| Περιεχόμενα .....  | 11 |
| Λίστα Εικόνων.....   | 13 |
| Λίστα Σχημάτων.....  | 16 |
| Λίστα Πινάκων .....  | 17 |
| ΚΕΦΑΛΑΙΟ 1 <sup>ο</sup> - ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΗΜΑΤΟΣ ΜΕ ΤΕΧΝΙΚΕΣ ΜΗ ΓΡΑΜΜΙΚΗΣ<br>ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ..... | 18 |
| 1.1 Εισαγωγή.....  | 18 |
| 1.2 Αναπαράσταση σήματος με τη βοήθεια διανυσματικών χώρων και στάθμες<br>διανυσμάτων .....      | 19 |
| 1.3 Αναπαράσταση σήματος με τη χρήση βάσεων .....  | 21 |
| 1.4 Αραιά μοντέλα κωδικοποίησης .....  | 23 |
| 1.5 Μοναδικότητα της αραιότερης λύσης .....  | 25 |
| ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> – OMP ΑΛΓΟΡΙΘΜΟΣ .....   | 29 |
| 2.1 Εισαγωγή στον αλγόριθμο OMP .....  | 29 |
| 2. Ο 2D μετασχηματισμός DCT στο υπερπλήρες λεξικό .....  | 31 |
| 2.3 Αποθρομβοποίηση εικόνας και αλγόριθμος OMP .....   | 32 |
| ΚΕΦΑΛΑΙΟ 3 <sup>ο</sup> - Ο k-SVD ΑΛΓΟΡΙΘΜΟΣ ΚΑΙ Η ΑΡΑΙΗ ΑΝΑΠΑΡΑΣΤΑΣΗ.....                       | 34 |
| 3.1 Εκμάθηση λεξικού .....   | 34 |
| 3.2. Γενική περιγραφή του αλγορίθμου (k-SVD) .....   | 38 |
| ΚΕΦΑΛΑΙΟ 4 <sup>ο</sup> - ΑΠΟΘΡΥΒΟΠΟΙΗΣΗ ΕΙΚΟΝΑΣ ΜΕ ΧΡΗΣΗ k-SVD .....                            | 43 |

|   |     |
|---|-----|
| 4.1 Γενικά .....  | 43  |
| 4.2. Μέθοδος των επικαλυπτόμενων τμημάτων .....           | 43  |
| 4.3 Αποθορυβοποίηση σήματος με θόρυβο Gaussian .....      | 44  |
| 4.4. Αφαίρεση ομοιόμορφου θορύβου από εικόνα .....        | 58  |
| 4.5. Αποθορυβοποίηση σήματος με θόρυβο salt n pepper..... | 71  |
| Κεφάλαιο 5° – ΑΝΑΚΑΤΑΣΚΕΥΗ ΤΜΗΜΑΤΩΝ ΕΙΚΟΝΑΣ .....         | 90  |
| 5.1 Γενικά .....  | 90  |
| 5.2 Ελλιπή δεδομένα εικόνων – αφαίρεση γραμμάτων .....    | 91  |
| Κεφάλαιο 6° - ΣΥΜΠΕΡΑΣΜΑΤΑ .....                          | 98  |
| Βιβλιογραφία.....   | 100 |

## Λίστα Εικόνων

|  |           |
|--|-----------|
| Εικόνα 4.1.a Image_1 με Block size 8, dict size 256 & sigma 10,20,3            | Error!    |
| <b>Bookmark not defined.....</b>   | <b>46</b> |
| Εικόνα 4.1.b Image_1 με Block size 8, dict size 256 & sigma 10,20,30.....      | 47        |
| Εικόνα 4.2.a Image_2 με Block size 8, dict size 256 & sigma 10,20,30.....      | 48        |
| Εικόνα 4.2.b Image_2 με Block size 8, dict size 256 & sigma 10,20,30.....      | 49        |
| Εικόνα 4.3.a Image_3 με Block size 8, dict size 256 & sigma 10,20,30.....      | 50        |
| Εικόνα 4.3.b Image_3 με Block size 8, dict size 256 & sigma 10,20,30.....      | 51        |
| Εικόνα 4.4.a Image_4 με Block size 8, dict size 256 & sigma 10,20,30.....      | 52        |
| Εικόνα 4.4.b Image_4 με Block size 8, dict size 256 & sigma 10,20,30.....      | 53        |
| Εικόνα 4.5.a Image_5 με Block size 8, dict size 256 & sigma 10,20,30.....      | 54        |
| Εικόνα 4.5.b Image_5 με Block size 8, dict size 256 & sigma 10,20,30.....      | 55        |
| Εικόνα 4.6. Image_1 με Block size 8, dict size 128 & sigma 30.....             | 56        |
| Εικόνα 4.7 Image_5 με Block size 4 dict size 128 & sigma 40.....               | 56        |
| Εικόνα 4.8 Image_3 με Block size 4, dict size 128 & sigma 10.....              | 56        |
| Εικόνα 4.9.a Image_1 με Block size 8, dict size 512 & sigma 10,20,30.....      | 59        |
| Εικόνα 4.9.b Image_1 με Block size 8, dict size 512 & sigma 10,20,30.....      | 60        |
| Εικόνα 4.10.a Image_2 με Block size 8, dict size 512 & sigma 10,20,30.....     | 61        |
| Εικόνα 4.10.b Image_2 με Block size 8, dict size 512 & sigma 10,20,30.....     | 62        |
| Εικόνα 4.11.a Image_3 με Block size 8, dict size 512 & sigma 10,20,30.....     | 63        |
| Εικόνα 4.11.b Image_3 με Block size 8, dict size 512 & sigma 10,20,30.....     | 64        |
| Εικόνα 4.12.a Image_4 με Block size 8, dict size 512 & sigma 10,20,30.....     | 65        |
| Εικόνα 4.12.b Image_4 με Block size 8, dict size 512 & sigma 10,20,30.....     | 66        |
| Εικόνα 4.13.a Image_5 με Block size 8, dict size 512 & sigma 10,20,3.....      | 67        |
| Εικόνα 4.13.b Image_5 με Block size 8, dict size 512 & sigma 10,20,3.....      | 68        |
| Εικόνα 4.14 Image_1 με Block size 8, sigma 20& dict size 64,128,256,512.....   | 69        |
| Εικόνα 4.15 Image_2 με Block size 8, sigma 20 & dict size 64,128,256,512 ..... | 70        |
| Εικόνα 4.16.a Image_1 με Block size 8, dict size 128 & sigma 10,20,30.....     | 71        |
| Εικόνα 4.16.b Image_1 με Block size 8, dict size 128 & sigma 10,20,30.....     | 72        |

|  |    |
|--|----|
| Εικόνα 4.17.a Image_2 με Block size 8, dict size 128 & sigma 10,20,30.....   | 73 |
| Εικόνα 4.17.b Image_2 με Block size 8, dict size 128 & sigma 10,20,30.....   | 74 |
| Εικόνα 4.18.a Image_3 με Block size 8, dict size 128 & sigma 10,20,30.....   | 75 |
| Εικόνα 4.18.b Image_3 με Block size 8, dict size 128 & sigma 10,20,30.....   | 76 |
| Εικόνα 4.19.a Image_4 με Block size 8, dict size 128 & sigma 10,20,30.....   | 77 |
| Εικόνα 4.19.b Image_4 με Block size 8, dict size 128 & sigma 10,20,30.....   | 78 |
| Εικόνα 4.20.a Image_5 με Block size 8, dict size 128 & sigma 10,20,30.....   | 79 |
| Εικόνα 4.20.b Image_5 με Block size 8, dict size 128 & sigma 10,20,30.....   | 80 |
| Εικόνα 4.21 Image_1 με Block size 8, dict size 256 & sigma 10.....   | 81 |
| Εικόνα 4.22 Image_4 με Block size 8, dict size 256 & sigma 10.....   | 81 |
| Εικόνα 4.23 Image_4 με Block size 8, dict size 156 & sigma 20.....   | 81 |
| Εικόνα 4.24 Image_4 με Block size 8, dict size 256 & sigma 30.....   | 81 |
| Εικόνα 4.25 Image_5 με Block size 8, dict size 256 & sigma 40.....   | 82 |
| Εικόνα 5.1 Image_6 Inpainting με k-SVD σε έγχρωμη εικόνα (αριστερά αρχική εικόνα, δεξιά αποτέλεσμα αλγορίθμου).....                                  | 92 |
| Εικόνα 5.2 Image_1 Inpainting με k-SVD σε grayscale εικόνα (αριστερά αρχική εικόνα, δεξιά αποτελέσματα αλγορίθμου).....                              | 92 |
| Εικόνα 5.3 Image_7 Inpainting με k-SVD σε έγχρωμη εικόνα (αριστερά αρχική εικ. με γράμματα, δεξιά αποτελέσματα αλγορίθμου) bb=8, overlap=3, j=3..... | 93 |
| Εικόνα 5.4 Image_7 Inpainting με k-SVD σε έγχρωμη εικόνα (αριστερά αρχική εικ. με γράμματα, δεξιά αποτελέσματα αλγορίθμου) bb=8, overlap=3, j=3..... | 93 |
| Εικόνα 5.5 Image_7 Inpainting με k-SVD σε έγχρωμη εικόνα (αριστερά αρχική εικ. με γράμματα, δεξιά αποτελέσματα αλγορίθμου) bb=8, overlap=2, j=2..... | 93 |
| Εικόνα 5.6 Image_6 Inpainting με k-SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων, bb=8, overlap=3, j=3.....  | 94 |
| Εικόνα 5.7 Image_6 Inpainting με k-SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων bb=8, overlap=3, j=3.....   | 94 |
| Εικόνα 5.8 Image_6 Inpainting με k-SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων bb=8, overlap=3, j=3.....   | 94 |
| Εικόνα 5.9 Image_6 Inpainting με k-SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων bb=8, overlap=2, j=2.....   | 94 |

|   |    |
|---|----|
| Εικόνα 5.10 Image_1 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=3,j=3$ ..... | 95 |
| Εικόνα 5.11 Image_2 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=3,j=3$ ..... | 95 |
| Εικόνα 5.12 Image_3 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=2,j=2$ ..... | 95 |
| Εικόνα 5.13 Image_4 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=3,j=3$ ..... | 96 |
| Εικόνα 5.14 Image_4 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=2,j=2$ ..... | 96 |
| Εικόνα 5.15 Image_5 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=3,j=3$ ..... | 96 |
| Εικόνα 5.16 Image_5 Inpainting με k-SVD σε grayscale εικόνα αφαίρεση<br>τμημάτων $bb=8,overlap=3,j=3$ ..... | 97 |

## Λίστα Σχημάτων

|  |    |
|--|----|
| Σχήμα 1: Ο αλγόριθμος OMP.....   | 30 |
| Σχήμα 2: 2D-DCT υπερπλήρες λεξικό.....   | 32 |
| Σχήμα 3: Το πρόβλημα Compressed sensing (CS).....  | 34 |
| Σχήμα 4: Το πρόβλημα Dictionary Learning (DL).....   | 35 |
| Σχήμα 5: Ο αλγόριθμος k-SVD.....   | 41 |
| Σχήμα 6: Κατανομή gauss για $\sigma = 20$ .....  | 45 |
| Σχήμα 7: Κατανομή με θόρυβο ομοιόμορφης κατανομής και $\sigma = 20$ .....                    | 58 |
| Σχήμα 8: Image_1 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 10$ .....  | 85 |
| Σχήμα 9: Image_1 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 20$ .....  | 85 |
| Σχήμα 10: Image_2 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 10$ ..... | 86 |
| Σχήμα 11: Image_2 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 20$ ..... | 86 |
| Σχήμα 12: Image_3 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 30$ ..... | 87 |
| Σχήμα 13: Image_4 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 40$ ..... | 87 |
| Σχήμα 14: Image_5 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & $\sigma = 10$ ..... | 88 |
| Σχήμα 15: Σύγκριση 5 εικόνων με θόρυβο Gauss & $\sigma = 10$ .....                           | 88 |
| Σχήμα 16: Σύγκριση 5 εικόνων με θόρυβο ομοιόμορφης κατανομής & $\sigma = 20$ .....           | 89 |
| Σχήμα 17: Σύγκριση 5 εικόνων με θόρυβο salt and pepper & $\sigma = 30$ .....                 | 89 |



## Λίστα Πινάκων

|  |    |
|--|----|
| Πίνακας 1 PSNR image_1 3 type noise & sigma 10,20,30,40..... | 82 |
| Πίνακας 2 PSNR για όλα τα image με block size 4,8.....       | 83 |
| Πίνακας 3 PSNR για όλα τα image με dict size 256,512.....    | 84 |

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> - ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΗΜΑΤΟΣ ΜΕ ΤΕΧΝΙΚΕΣ ΜΗ ΓΡΑΜΜΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

### 1.1 Εισαγωγή

Τις δύο τελευταίες δεκαετίες στον τομέα της επεξεργασίας εικόνας έχει παρατηρηθεί ιδιαίτερο ενδιαφέρον στην κωδικοποίηση σημάτων με χρήση αραιής αναπαράστασης. [1]. Στην περίπτωση κατά την οποία έχουμε μεγάλο όγκο από δεδομένα χρησιμοποιούμε την τεχνική της αναπαράστασης των σημάτων με χρήση βάσεων ή λεξικό. Η τεχνική αυτή στηρίζεται στις αναπαραστάσεις πληθώρας σημάτων από μικρά διανύσματα παράλληλα με ένα κατάλληλο μητρώο το οποίο λειτουργεί σαν λεξικό ή βάση για την καθολική περιγραφή του σήματος [1]. Στη συνέχεια η ανακατασκευή του σήματος στηρίζεται σε μετρήσεις που βασίζονται σε μη γραμμικές βελτιστοποιήσεις (nonlinear optimizations). Σε αυτή την εργασία θα μελετήσουμε τον αλγόριθμο εκμάθησης λεξικών k-SVD ο οποίος αποτελεί μία ευέλικτη διαδικασία που με την εκπαίδευση ενός υπερπλήρους λεξικού ταιριάζει καλύτερα σε ένα σύνολο δεδομένων σημάτων. Έτσι η ανάπτυξη ενός αλγορίθμου που ελαχιστοποιεί ή αφαιρεί τα θορυβώδη εικονοστοιχεία μίας εικόνας παράγει ως τελικό αποτέλεσμα μία εικόνα σχεδόν χωρίς καμία παραμόρφωση.

Η παρούσα διπλωματική εργασία δομείται σε 6 κεφάλαια. Στο πρώτο κεφάλαιο ασχολούμαστε με την αναπαράσταση σημάτων με χρήση διανυσμάτων και ιδιαίτερα των αραιών διανυσμάτων. Στο δεύτερο κεφάλαιο μελετάμε τον αλγόριθμο OMP εστιάζοντας κυρίως στη λειτουργία του κατά την αποθρομβοποίηση μιας εικόνας. Στη συνέχεια μελετάμε τον αλγόριθμο k-SVD (τρίτο κεφάλαιο), ενώ στο τέταρτο κεφάλαιο εφαρμόζουμε τον συγκεκριμένο αλγόριθμο σε πλήθος εικόνων που έχουν υποστεί θορυβοποίηση μελετώντας και μετρώντας σε κάθε περίπτωση την αποθρομβοποίηση που έχει επιτευχθεί.

Στο πέμπτο κεφάλαιο με την χρήση του k-SVD προχωράμε στην ανακατασκευή αλλοιωμένων τμημάτων σε εικόνες ή ελλειπών περιοχών σε αυτές. Τέλος στο έκτο και τελευταίο κεφάλαιο έχουμε συγκεντρώσει τα συμπεράσματα και τις μελλοντικές

επεκτάσεις που δύναται να έχει το αντικείμενο του θέματος που εκπονείται στην εργασία μας.

## 1.2 Αναπαράσταση σήματος με τη βοήθεια διανυσματικών χώρων και στάθμες διανυσμάτων

Για την αναπαράσταση σημάτων στον τομέα της επεξεργασίας εικόνας χρησιμοποιούμε τους διανυσματικούς χώρους. Η παρατήρηση ότι τα φυσικά συστήματα υπακούουν στο νόμο της γραμμικότητας έχει οδηγήσει στην περιγραφή τους με χρήση των διανυσματικών χώρων. Για τον μηχανικό επεξεργασίας εικόνας ο διανυσματικός χώρος είναι μία σημαντική εργαλειοθήκη, αφού με τη χρήση τους μπορούν να οπτικοποιηθούν τα αποτελέσματα της επεξεργασίας εικόνων. Οι αποστάσεις όπως επίσης οι στάθμες και οι γωνίες των διανυσμάτων είναι χρήσιμα μετρικά μεγέθη για την αναπαράσταση και επεξεργασία εικόνας. Από αυτές οι νόρμες διανυσμάτων ( $\ell_p$ ) είναι από τις πιο εύχρηστες μετρικές παραμέτρους για το μέγεθος ενός σήματος. [2]

**Ορισμός 1.1.** Θεωρούμε ένα διάνυσμα  $x$  το οποίο αποτελείται από  $n$  στοιχεία  $x_i$ ,  $i=1,2,\dots,n$ , και ορίζεται στον Ευκλείδειο χώρο  $R^n$ . Καλούμε την  $\ell_p$  ως νόρμα (norm) του διανύσματος  $x$  για κάποιο  $p \in \{1, \infty\}$  την ποσότητα

$$\ell_p = \|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, p \in [1, \infty) \quad (1)$$

Επιπλέον, όταν  $p \rightarrow \infty$  ορίζεται η  $\ell_\infty$  στάθμη διανύσματος ως

$$\ell_\infty = \|x\|_\infty = \max_{i=1,2,\dots,n} |x_i| \quad (2)$$

Για τις στάθμες  $\ell_p$  των διανυσμάτων ισχύουν οι παρακάτω ιδιότητες:

1.  $\|x\|_p \geq 0$
2.  $\|x\|_p = 0 \Leftrightarrow x = 0$
3.  $\|ax\|_p = |a| \|x\|_p \quad \forall a \in R$
4.  $\|x_1 + x_2\|_p \leq \|x_1\|_p + \|x_2\|_p$

όταν  $p=1$  η  $\ell_1$  στάθμη του διανύσματος  $x$  αποτελεί το άθροισμα των απόλυτων τιμών των συνιστωσών του διανύσματος δηλ.,

$$\ell_1 = \|x\|_0 = \sum_{i=1}^n |x_i| \quad (3)$$

Αντίστοιχα όταν  $p=2$  η  $\ell_2$  στάθμη ταυτίζεται με την Ευκλείδεια στάθμη.

Οι  $\ell_p$  στάθμες των διανυσμάτων μπορούν να επεκταθούν και για αριθμούς που ανήκουν στο διάστημα  $p \in (0,1)$ . Σε τέτοιες περιπτώσεις οι στάθμες  $\ell_p$  που δημιουργούνται δεν ικανοποιούν την 4<sup>η</sup> κατά σειρά ιδιότητα των συναρτήσεων νόρμας (τριγωνική ανισότητα) και για αυτό τον όρο «στάθμη» σε αυτές τις περιπτώσεις τον χρησιμοποιούμε κατά κόρον.

Όταν  $p \rightarrow 0$  έχουμε την «στάθμη»  $\ell_0$  για ένα διάνυσμα  $x$  που ορίζεται ως εξής:

$$\ell_0 = \|x\|_0 = \sum_{i=1}^n X_{(0,\infty)}(|x_i|), \quad (4)$$

όπου η συνάρτηση  $X_A$  είναι η χαρακτηριστική συνάρτηση ενός συνόλου  $A$  και για ένα στοιχείο  $x$  ορίζεται ως εξής:

$$X_A(x) := \begin{cases} 0, & x \in A \\ 1, & x \notin A \end{cases} \quad (5)$$

Οπότε, η μηδενική στάθμη ενός διανύσματος  $x$  περιέχει όλα εκείνα τα στοιχεία του  $x$  που είναι μη μηδενικά. Η  $\ell_0$  «στάθμη» ικανοποιεί την ιδιότητα της τριγωνικής ανισότητας σε αντίθεση με τις «στάθμες» του διαστήματος  $(0,1)$ , αλλά δεν ικανοποιεί την 3<sup>η</sup> ιδιότητα, δηλαδή της θετικά ορισμένης ομογενούς συνάρτησης 1<sup>ου</sup> βαθμού.

Άμεση συνέπεια των παραπάνω είναι ότι η προς τα πάνω μεταβολή της τιμής ενός στοιχείου με τη προϋπόθεση όχι  $|x| > 1$  και δεδομένου  $p \geq 1$ , έπεται σημαντικότερη συνεισφορά αυτού του στοιχείου στην τιμή της στάθμης και

αντίστροφα. Για κάθε στάθμη  $\ell_p$ , όπου για την οποία ισχύει  $p \geq 1$ , η μοναδική στάθμη η οποία διατηρεί κάποια μερική αναλογική συνεισφορά στο μέγεθος συνεισφοράς ενός στοιχείου στην τιμή της στάθμης είναι η  $\ell_1$  (δηλ. παίρνει υπόψη της και τα «μικρά στοιχεία» του διανύσματος).

Συνοψίζοντας τα παραπάνω και σε συνδυασμό ότι η στάθμη  $\ell_1$  σε μία επαναληπτική διαδικασία θα διατηρήσει εκείνα τα στοιχεία του διανύσματος παραμέτρων τα οποία βελτιστοποιούν την προσαρμογή (fitting) των δεδομένων στο πρόβλημα μηδενίζοντας παράλληλα τα υπόλοιπα στοιχεία (παραμέτρους) προκύπτει ότι η αντικατάσταση της στάθμης  $\ell_2$  από την  $\ell_1$  θα παράγει ένα διάνυσμα με πολλά μηδενικά στοιχεία. Η παραπάνω παρατήρηση μειώνει τη διάσταση ενός προβλήματος συμπιέζοντας κατά αυτό τον τρόπο το μέγεθος του (συμπιεσμένη μορφή).

### 1.3 Αναπαράσταση σήματος με τη χρήση βάσεων

Η αύξηση των δεδομένων ενός σήματος οδηγεί στην κωδικοποίηση του με τη χρήση διανυσματικών βάσεων.

**Ορισμός 1.2.** Σε ένα διανυσματικό χώρο του  $R^n$ , ορίζουμε βάση (basis) το σύνολο των γραμμικά ανεξάρτητων διανυσμάτων  $\varphi_i$  για τα οποία κάθε διάνυσμα  $x$  μπορεί να εκφραστεί από την σχέση

$$x = \sum_{i=1}^n c_i \varphi_i \quad (6)$$

Στο χώρο της επεξεργασίας σήματος αλλά και της μηχανικής μάθησης τη βάση την καλούμε λεξικό και τα διανύσματα βάσης άτομα. Θεωρώντας ότι έχουμε ένα μητρώο  $\Phi$  ( $n \times n$ ) με στήλες τα  $\varphi_i$  και  $c$  το  $n \times 1$  διάνυσμα με στοιχεία  $c_i$  η παραπάνω εξίσωση (6) μπορεί να γραφεί ως:

$$x = \Phi c \quad (7)$$

Οι ορθοκανονικές βάσεις είναι μια ειδική κατηγορία βάσεων. Η βάση για την οποία είναι  $\{\varphi_i\}_{i=1}^n$  ονομάζεται ορθοκανονική όταν ισχύει:

$$\varphi_i \varphi_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (8)$$

ή εναλλακτικά

$$\Phi^T \Phi = I \quad (9)$$

Άρα για μία βάση που είναι ορθοκανονική θα ισχύει:

$$c = \Phi^T x \quad (10)$$

Οι βάσεις μπορούν να επεκταθούν και για σύνολα με γραμμικά εξαρτημένα διανύσματα. Όταν έχουμε μία τέτοια περίπτωση η έννοια της βάσης εναλλάσσεται και την θέση παίρνει η έννοια του πλαισίου (frame).

**Ορισμός 1.3.** Θεωρούμε χώρο διάστασης  $d$  πραγματικών αριθμών  $R^d$  ένα σύνολο γραμμικά ανεξάρτητων διανυσμάτων  $\{\varphi_i\}_{i=1}^n$  με  $n > d$ . Το σύνολο αυτό το ονομάζουμε πλαίσιο (frame) αν για το μητρώο  $\Phi_{d \times n}$  με στήλες τα διανύσματα  $\varphi_i$  και για κάθε διάνυσμα  $x$  του χώρου  $R^d$  υπάρχουν πραγματικές σταθερές  $A$  και  $B$  με  $0 < A \leq B$  έτσι ώστε

$$A \|x\|_2^2 \leq \|\Phi^T x\|_2^2 \leq B \|x\|_2^2 \quad (11)$$

Πολλές φορές τον όρο πλαίσιο στην επιστήμη της επεξεργασίας σήματος τον συναντάμε ως υπερπλήρες λεξικό (overcomplete dictionary).

Η αναπαράσταση σημάτων με χρήση πλαισίων δίνει καλύτερα αποτελέσματα από αυτή της χρήσης βάσεων και αυτό συμβαίνει λόγω της πληθώρας ατόμων που μπορούμε να χρησιμοποιήσουμε ως μονάδες αναπαράστασης. Στην περίπτωση που έχουμε πλαίσια η εξίσωση (7) με άγνωστο αριθμό αυτό του διανύσματος  $c$  θεωρείτε ως ένα γραμμικό σύστημα με  $n$  αγνώστους και  $d$  εξισώσεις όπου  $d < n$  το οποίο έχει άπειρη ομάδα λύσεων. Αυτό που περιγράψαμε είναι η βασική ιδέα της αναπαράστασης ενός σήματος με τη χρήση διανύσματος με ελάχιστους συντελεστές διάφορους του μηδενός και ενός υπερπλήρους λεξικού.

## 1.4 Αραιά μοντέλα κωδικοποίησης

Μπορούμε να αναπαραστήσουμε ένα σήμα πολύ καλύτερα ως γραμμικό συνδυασμό όταν χρησιμοποιήσουμε έναν μικρό αριθμό από τα διανύσματα μιας βάσης.

**Ορισμός 4.4.** Έστω ότι για να αναπαραστήσουμε ένα σήμα χρησιμοποιούμε ένα διάνυσμα  $x$  τότε αυτό είναι  $k$ -αραιό ( $k$ -sparse) όταν το πλήθος των στοιχείων που είναι διάφορα του μηδενός ισούται το πολύ  $k$  δηλαδή ισχύει:

$$\|x\|_0 \leq k \quad (12)$$

Αυτό που προσπαθούμε να επιτύχουμε δεν είναι η εύρεση αραιών σημάτων αλλά ο στόχος μας είναι να προσεγγίσουμε αυτά τα σήματα με την χρήση των λεξικών που αναφέρθηκαν παραπάνω. Έτσι θεωρούμε ένα σήμα  $x$  ότι είναι  $k$ -αραιό όταν έχουμε την δυνατότητα να το αναπαραστήσουμε με μία εξίσωση της μορφής (7) όπου και το διάνυσμα  $c$  είναι και αυτό  $k$ -αραιό.

Έστω ότι θέλουμε να λύσουμε το ακόλουθο σύστημα της γραμμικής εξίσωσης:

$$y = X\theta \quad (13)$$

που περιγράφει ένα συνηθισμένο μοντέλο εισόδων στην επεξεργασία σημάτων. Έστω ότι  $y$  είναι το  $N \times 1$  διάνυσμα των εισόδων,  $X$  είναι το  $N \times I$  μητρώο εισόδου, για το οποίο υποθέτουμε ότι έχει βαθμό  $\text{rank}(X) = N$ , και  $\theta$  είναι  $I \times 1$  άγνωστο διάνυσμα παραμέτρων. Λόγω του ότι  $N < I$  το προαναφερόμενο γραμμικό σύστημα έχει μικρότερο αριθμό εξισώσεων από τους αγνώστους (underdetermined system) και μπορεί να έχει άπειρες λύσεις. Αν θέλουμε να δούμε το πρόβλημα αυτό από γεωμετρική άποψη, το σύνολο των πιο πιθανών λύσεων του βρίσκονται στην τομή των  $N$  πολλαπλών επιπέδων του  $I$ -διάστατου χώρου, δηλαδή

$$y_n = x_n^T \theta \quad n=1,2,\dots,N \quad \theta \in R^I \quad (14)$$

Έχοντας ως δεδομένο ότι το μητρώο  $X$  έχει βαθμό  $N$  τα  $N$  πολλαπλά επίπεδα που περιγράφονται από την εξίσωση (14) τέμνοντας την και η τομή τους είναι ένα πολλαπλό επίπεδο διάστασης  $I - N$ . Άρα οι πιθανές λύσεις του συστήματος (14) στο

σύνολο τους βρίσκονται σε ένα πολλαπλό επίπεδο με διάσταση  $I - N$ . όσον αφορά την αλγεβρική ερμηνεία των λύσεων αυτών χρειαζόμαστε τον ακόλουθο ορισμό:

**Ορισμός 1.5.** Υποθέτουμε ότι μας δίνεται το  $N \times I$  μητρώο  $X$ . Ορίζουμε ως null space τον γραμμικό υπόχωρο που ορίζουν τα διανύσματα  $z$  και συμβολίζουμε με  $null(X)$  για τα οποία ισχύει η εξίσωση:

$$null(X) = \{z \in R^I : Xz = 0\} \quad (15)$$

Θεωρούμε ότι  $\Theta$  είναι το σύνολο των λύσεων και ότι  $\theta_0$  είναι μια λύση του συστήματος (14). Άρα  $\forall \theta \in \Theta$  θα ισχύει:

$$X(\theta - \theta_0) = 0 \Rightarrow \theta - \theta_0 \in null(X) \Rightarrow \Theta = \theta_0 + null(X) \quad (16)$$

Το null space του  $N \times I$ μητρώο  $X, N$  βαθμού, όταν  $I > N$  θεωρείτε ως ένας γραμμικός υπόχωρος διάστασης  $I - N$ . Το να επιλέξουμε μια συγκεκριμένη λύση του συστήματος ή διαφορετικά ένα συγκεκριμένο σημείο του υποχώρου  $\Theta$  χρειάζεται επιπρόσθετους περιορισμούς. Ένας από τους περιορισμούς που χρησιμοποιούμε είναι η στάθμη ενός διανύσματος. Έτσι έχουμε στόχο να εντοπίσουμε μια λύση του συστήματος με την οποία ικανοποιείται το κριτήριο ελαχιστοποίησης μιας συγκεκριμένης στάθμης. Το πρόβλημα βελτιστοποίησης που προκύπτει σε μια γενική μορφή μπορεί να γραφεί ως:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^I} \quad & \|\theta\|_p \\ \text{s.t.} \quad & y_n = x_n^T \theta \quad n=1,2,\dots,N \end{aligned} \quad (17)$$

Εφόσον ο στόχος μας είναι να έχουμε ένα διάνυσμα όσο γίνεται πιο αραιό, τότε είναι προφανές ότι η στάθμη που πρέπει να χρησιμοποιήσουμε είναι η  $\ell_0$ . Αν ελαχιστοποιήσουμε τη στάθμη  $\ell_0$  εξασφαλίζουμε ότι η λύση που μας προκύπτει η οποία θα είναι με κριτήριο το πλήθος των μηδενικών στοιχείων, θα μας δώσει το βέλτιστο διάνυσμα παραμέτρων  $\theta$ . Αυτό το πρόβλημα βελτιστοποίησης είναι πολύ δύσκολο να επιλυθεί και είναι απαραίτητη η μέθοδος της εξαντλητικής αναζήτησης



(NP- hard problem) για την επίλυση του. Η μοναδικότητα της λύσης του προβλήματος θα αναλυθεί σε επόμενο κεφάλαιο.

### 1.5 Μοναδικότητα της αραιότερης λύσης

Για να μπορέσουμε να εξασφαλίσουμε ότι θα επιτύχουμε τις κατάλληλες συνθήκες που θα μας δώσουν την μοναδικότητα στη λύση του προβλήματος βελτιστοποίησης που έχουμε στην εξίσωση (17) είναι απαραίτητος ο ακόλουθος ορισμός:

**Ορισμός 1.6.** Έχοντας το  $N \times I$  μητρώο  $X$  όπου  $I \geq N$  και βαθμό  $N$ . Θεωρούμε ως  $\text{spark}$  του μητρώου  $X$  έναν αριθμό που ισούται με το μικρότερο από το πλήθος των γραμμικά ανεξάρτητων στηλών του μητρώου  $X$  και που το συμβολίζουμε με  $\text{spark}(X)$ .

Σύμφωνα με τον Ορισμό 1.6. οποιαδήποτε από τις  $m$  στήλες του μητρώου  $X$  με  $m < \text{spark}(X)$  είναι γραμμικά ανεξάρτητη. Ένας αποδοτικός αλγόριθμος δεν μπορεί να υπολογίσει την εύρεση του  $\text{spark}$  ενός μητρώου, γι' αυτό απαιτείται η μέθοδος εξαντλητικής αναζήτησης.

Το Λήμμα που ακολουθεί συνδέει το  $\text{spark}$  ενός μητρώου με το πρόβλημα της βελτιστοποίησης (17).

**Λήμμα 1.1.** Αν θεωρήσουμε ότι για το σύστημα γραμμικών εξισώσεων (13) υπάρχει μία λύση  $\theta_0$  για την οποία ισχύει

$$\|\theta_0\|_0 < \frac{1}{2} \text{spark}(X) \quad (18)$$

Τότε το  $\theta_0$  θα αντιστοιχεί στη λύση του προβλήματος βελτιστοποίησης (17) δηλαδή θα είναι η αραιότερη λύση του συστήματος.

Η αναζήτηση και η εύρεση του  $\text{spark}$  από ένα μητρώο είναι ένα πρόβλημα που χαρακτηρίζεται δύσκολο, έχει ανακαλυφθεί μια άλλη μετρική που έχει σχέση με την εύρεση του ελαχιστοποιητή  $\ell_0$  για το πρόβλημα (17). Η αμοιβαία συνοχή (mutual coherence) ενός μητρώου είναι η μετρική αυτή.

**Ορισμός 1.7.** Έστω το  $N \times I$  μητρώο  $X$  για το οποίο έχουμε τις στήλες  $x_i$ ,  $i=1,2,\dots,I$ . Θέτουμε ως αμοιβαία συνοχή (mutual coherence) του μητρώου  $X$  και το συμβολίζουμε με  $\mu(X)$  την ακόλουθη ισότητα:

$$\mu(X) = \max_{1 \leq i \leq j \leq I} \frac{|x_i^T x_j|}{\|x_i\| \|x_j\|} \quad (19)$$

Για την αμοιβαία συνοχή (mutual coherence) ενός μητρώου  $X$  ισχύει πάντα  $0 \leq \mu(X) \leq 1$ . Αντίστοιχα για ένα  $N \times I$  μητρώο  $X$ , όταν  $I > N$  ισχύει πάντα

$$\sqrt{\frac{I-N}{N(I-1)}} \leq \mu(X) \leq 1 \quad (20)$$

Το παραπάνω κάτω φράγμα είναι γνωστό και ως Welch bound και για  $I \square N$  ισχύει  $\mu(X) \geq \frac{1}{\sqrt{N}}$ . Το επόμενο Λήμμα συνδέει το spark με την αμοιβαία συνοχή (mutual coherence) ενός μητρώου.

**Λήμμα 1.2.** Θεωρούμε ένα  $N \times I$  μητρώο  $X$ . Τότε θα ισχύει

$$\text{spark}(X) \geq 1 + \frac{1}{\mu(X)} \quad (21)$$

Στηριζόμενοι στο προηγούμενο Λήμμα και στο Λήμμα 1.1 προκύπτει άμεσα το επόμενο θεώρημα.

**Θεώρημα 1.1.** Εφόσον για το σύστημα γραμμικών εξισώσεων (13) διαθέτουμε μία λύση  $\theta_0$  για την οποία ισχύει

$$\|\theta_0\|_0 < \frac{1}{2} \text{spark}(X) \quad (22)$$

τότε το  $\theta_0$  αποτελεί την λύση του προβλήματος βελτιστοποίησης (17) δηλαδή είναι η αραιότερη λύση που μπορούμε να επιτύχουμε για το σύστημα. Έχοντας ως δεδομένο ότι η λύση του προβλήματος βελτιστοποίησης (17) είναι ένα NP-hard πρόβλημα, το να βρούμε κάποια από τα φράγματα είναι ένας τρόπος αντιμετώπισης ενός τέτοιου προβλήματος. Ιδιαίτερα σημαντικό είναι το γεγονός ότι κάτω από ορισμένες συνθήκες είναι εφικτή η ελαχιστοποίηση της στάθμης  $\ell_1$  αντί για τη

στάθμη  $\ell_0$ , και αυτό μπορεί να συμβεί χωρίς να μεταβάλει την επιδιωκόμενη λύση που είναι να επιτύχουμε το αραιότερο διάνυσμα παραμέτρων.

**Θεώρημα 1.2.** Έστω ότι έχουμε ένα γραμμικό σύστημα εξισώσεων όπως το (13) όπου  $X$  ένα μητρώο  $N \times I$ , με  $I > N$  και βαθμό  $N$ . Το σύστημα έχει λύση εάν ικανοποιείται η συνθήκη

$$\|\theta\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(X)}\right) \quad (23)$$

τότε αυτή θα είναι και η λύση του προβλήματος βελτιστοποίησης (17) ακόμη και αν αντικαταστήσουμε τη στάθμη  $\ell_0$  και ελαχιστοποιήσουμε τη στάθμη  $\ell_1$ . Το θεώρημα αυτό διατυπώθηκε το 2003 ξεχωριστά από τους Dono και Grib και έγινε το θεωρητικό υπόβαθρο των μέχρι τότε εμπειρικών αποτελεσμάτων δηλαδή ότι σε αρκετές περιπτώσεις ο  $\ell_0$  ελαχιστοποιητής είναι ίδιος με τον  $\ell_1$  ελαχιστοποιητή.

Η ισοδυναμία των  $\ell_0$  και  $\ell_1$  θεμελιώνεται με την Ιδιότητα Περιορισμένης Ισομετρίας η οποία δημοσιεύθηκε το 2006 από τον Candes [3].

**Ορισμός 1.8** Η Ιδιότητα Περιορισμένης Ισομετρίας (Restricted Isometry Property – RIP). Έστω ότι για κάθε ακέραιο  $k = 1, 2, \dots, n$  και για ένα  $N \times I$  μητρώο  $X$  ορίζεται η παράμετρος ισομετρίας (isometry parameter)  $\delta_k$  έτσι ώστε να ισχύει η συνθήκη

$$(1 - \delta_k) \|\theta\|_2^2 \leq \|X\theta\|_2^2 \leq (1 + \delta_k) \|\theta\|_2^2 \quad (24)$$

για όλα τα  $k$ -αραιά διανύσματα  $\theta$ .

**Θεώρημα 1.3.** Έστω το ακόλουθο πρόβλημα βελτιστοποίησης

$$\begin{aligned} \min_{\theta \in \mathbb{R}^I} \quad & \|\theta\|_1 \\ \text{s.t.} \quad & y_n = x_n^T \theta \quad n=1, 2, \dots, N \end{aligned} \quad (25)$$

Έστω ότι έχουμε την παράμετρο ισομετρίας  $\delta_{2k}$  και το μητρώο  $X = [x_1, x_2, \dots, x_n]$  για κάποιο  $k$  ισχύει  $\delta_{2k} < \sqrt{2} - 1$ . Τότε η λύση  $\theta^*$  του  $\ell_1$  ελαχιστοποιητή του

προβλήματος βελτιστοποίησης που μόλις είδαμε ικανοποιεί τις δύο επόμενες συνθήκες:

- $\|\theta - \theta^*\|_1 \leq C_0 \|\theta - \theta_k\|_1$
- $\|\theta - \theta^*\|_2 \leq C_0 k^{-\frac{1}{2}} \|\theta - \theta_k\|_1$

για κάποια σταθερά  $C_0$ . Καταλήγουμε ότι το διάνυσμα  $\theta$  είναι αυτό το αραιό διάνυσμα-στόχος του προβλήματος βελτιστοποίησης που αναζητούμε, ενώ το διάνυσμα  $\theta_k$  είναι το διάνυσμα που προκύπτει από το  $\theta$  αν διατηρήσουμε τα  $k$  μεγαλύτερα στοιχεία του ίδια και τα υπόλοιπα τα θέσουμε ίσα με το μηδέν.

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> – OMP ΑΛΓΟΡΙΘΜΟΣ

### 2.1 Εισαγωγή στον αλγόριθμο OMP

Ικανοποιητική λύση σε NP-hard προβλήματα δίνουν διαφόρων ειδών «άπληστοι» αλγόριθμοι (greedy algorithms), συνεπώς και για την ελαχιστοποίηση της  $\ell_0$  χρησιμοποιούνται τέτοιοι «άπληστοι» αλγόριθμοι οι οποίοι δημιουργούνται με βάση τις επαναληπτικές προσεγγίσεις των συντελεστών των σημάτων αραιής αναπαράστασης.

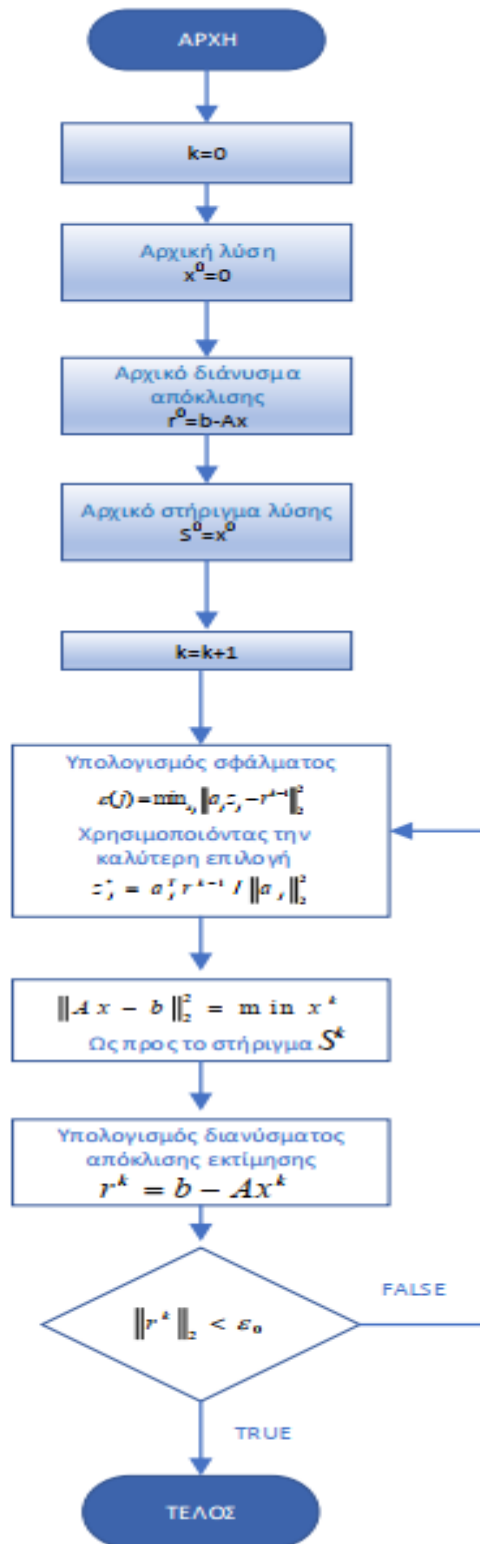
Ένας από τους πιο γνωστούς αλγορίθμους είναι ο OMP (Orthogonal Matching Pursuit) αλγόριθμος, που αναπτύχθηκε με βάση την basis pursuit μέθοδο που περιγράφεται στο [4], πρόταση των Pati, Rezaifar και Krishnaprasad το 1993 [5]. Ο αλγόριθμος αυτός χρησιμοποιείται σε ένα στάδιο του k-SVD που δημιουργείται και παίζει βασικό ρόλο στην εργασία μας. Η βελτιστοποίηση είναι ένα από τα πλέον γνωστά προβλήματα που προσπαθεί να λύσει ο OMP αλγόριθμος.

$$\min \|x\|_0 \text{ s.t. } Ax = b \quad (26)$$

Ο OMP αλγόριθμος αρχίζει έχοντας ως πρώτο στόχο να βρει τη στήλη του μητρώου  $A$  που συσχετίζεται περισσότερο με το σήμα  $b$ . Από την επανάληψη του βήματος αυτού προκύπτει η εκτίμηση της απόκλισης από το 1<sup>ο</sup> σήμα  $b$  και προσεγγίζουμε αυτό με το γινόμενο  $Ax$ . Οι μη μηδενικοί συντελεστές των αραιών διανυσμάτων  $x$  ή η ποσότητα που προσεγγίζει το σήμα  $b$  το γινόμενο  $Ax$ , δημιουργούν τα κριτήρια τερματισμού του αλγορίθμου.

Οι δύο φάσεις του αλγορίθμου OMP διακρίνονται έχοντας ως γνωστά αντικείμενα μελέτης τα εξής:

- 1) Το μητρώο  $A$
- 2) το διάνυσμα  $b$  και
- 3)  $\varepsilon_0$  το κατώφλι σφάλματος.



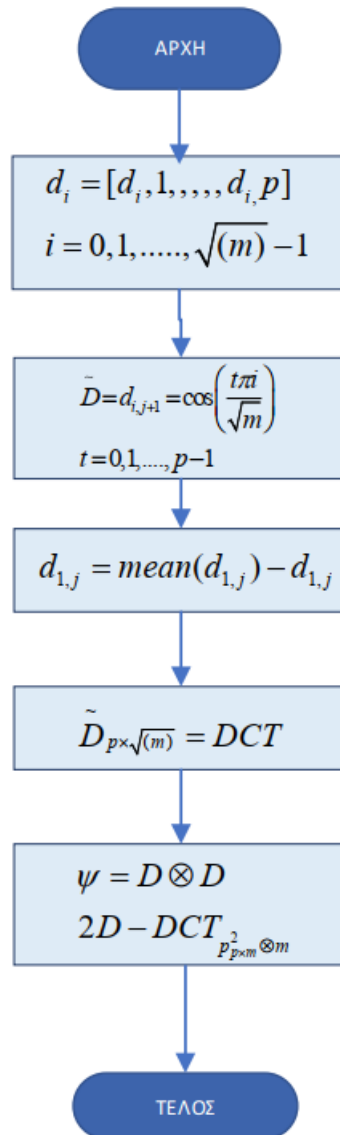
Σχήμα 1: Ο αλγόριθμος OMP

Ο αλγόριθμος OMP έχει βαθμό υπολογιστικής πολυπλοκότητας ίσο με  $O(k \log n)$ , όπου  $k$  μας δείχνει το πόσο αραίο είναι το διάνυσμα  $x$  και  $n$  οι διαστάσεις που έχει το σήμα  $b$ . Δοθέντος πως ο αλγόριθμος OMP είναι άπληστος δίνεται μεγάλη

πιθανότητα να βρίσκουμε την πιο επιτυχημένη  $k$ -αραιή αναπαράσταση του  $n$ -διάστατου σήματος  $b$ , με βήματα  $\leq k \log n$ . Πέραν του αλγορίθμου OMP, για να λύσουμε το συγκεκριμένο πρόβλημα χρησιμοποιούμε και έναν άλλο γνωστό αλγόριθμο LASSO (Least Absolute Shrinkage and Selection Operator), ο οποίος όμως δεν χρησιμοποιήθηκε από εμάς σε αυτή τη μελέτη.

## 2. Ο 2D μετασχηματισμός DCT στο υπερπλήρες λεξικό

Στην ψηφιακή επεξεργασία εικόνας συνήθως το λεξικό που χρησιμοποιείται για την αραιή αναπαράσταση ενός δισδιάστατου υπερπλήρους λεξικού είναι το 2D-DCT discrete cosine transform ( διακριτό μετασχηματισμό συνημιτόνου). Αυτό λεξικό αποτελείται από μητρώο διάστασης  $p \times \sqrt{m}$ ,  $p \times p$  ίσο με το μέγεθος των blocks-patches εκεί δηλαδή που διαιρείται η εικόνα μας και χρησιμοποιούμε την τεχνική των overlapped patches, μέθοδος που θα δούμε σε επόμενο κεφάλαιο. Ακόμα  $m$  θεωρούμε τον αριθμό των στοιχείων (atoms) που αποτελούν το υπερπλήρες λεξικό μας. Όταν έχουμε ένα υπερπλήρες λεξικό ισχύει  $m > p^2$ . Για να σχηματιστεί ένα 2D-DCT υπερπλήρες λεξικό πρέπει να προηγηθούν τα ακόλουθα βήματα [6] :



Σχήμα 2: 2D-DCT υπερπλήρη λεξικό

### 2.3 Αποθορυβοποίηση εικόνας και αλγόριθμος OMP

Η αποθορυβοποίηση είναι ένα σημαντικό έργο για την επεξεργασία εικόνας. Στο κεφάλαιο που προηγήθηκε, καταλήξαμε ότι ένας OMP αλγόριθμος έχοντας ως βάση ένα 2D-DCT υπερπλήρες λεξικό μπορεί να πραγματοποιήσει την αναπαράσταση μιας εικόνας έχοντας πιο λίγα δεδομένα και χωρίς μεγάλη μεταβολή στην ποιότητα της. Συμπεραίνουμε λοιπόν πως ένας πολύ καλός τρόπος να κάνουμε συμπίεση σε εικόνες είναι χρησιμοποιώντας τον αλγόριθμο OMP. Επίσης με αυτόν τον αλγόριθμο μπορούμε να καταφέρουμε και την αποθορυβοποίηση ψηφιακών εικόνων.



Έχοντας ως σήμα αρχικά μια εικόνα με θόρυβο, περιορίζουμε σε ένα ποσοστό τον θόρυβο αφού έχουμε αναπαραστήσει το μητρώο της εικόνας που επεξεργαζόμαστε σαν γινόμενο αραιών διανυσμάτων και με ένα σταθερό κατάλληλο λεξικό (fixed dictionary). Στην εργασία μας τέτοιο λεξικό είναι το 2D-DCT υπερπλήρες λεξικό. Η διαδικασία που ακολουθεί ο αλγόριθμος OMP για να απορρίψει κάποιο μέρος από τον θόρυβο είναι να εντοπίσει τα πιο «ταιριαστά - κοντινά» μέρη από το λεξικό ώστε να αναπαραστήσει το σήμα της αρχικής μας εικόνας.

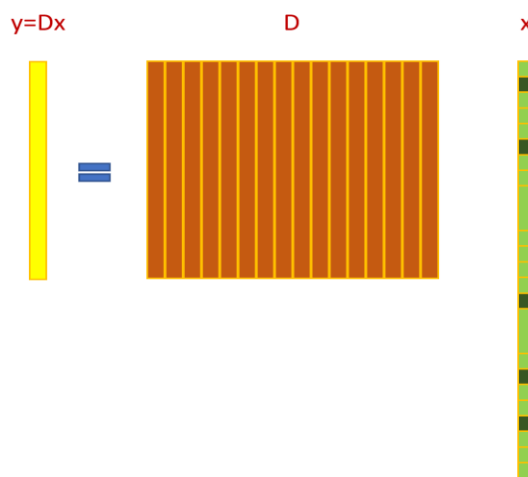
Η αποθορυβοποίηση ενός σήματος με την εφαρμογή του αλγορίθμου OMP επιτυγχάνεται με τη συμβολή ενός μικρού πλήθους συντελεστών από τα αραιά διανύσματα. Ενώ φαινομενικά η αύξηση του πλήθους των συντελεστών θα μπορούσε να οδηγήσει σε καλύτερα αποτελέσματα, τελικά φαίνεται ότι ο θόρυβος που περιέχεται στις εικόνες προσαρμόζεται καλύτερα στο προκύπτουν σήμα και τελικά δεν απορρίπτεται.

Με την μέθοδο της αραιής αναπαράστασης μιας εικόνας χρησιμοποιώντας ένα 2D-DCT υπερπλήρη λεξικό, αυξάνοντας τους συντελεστές του OMP αλγορίθμου οδηγούμαστε σε καλύτερα αποτελέσματα. Αντίθετα κάνοντας αποθορυβοποίηση σε μια εικόνα αυξάνοντας τους συντελεστές (από κάποιο σημείο και ύστερα) οδηγούμαστε στο να μην μπορούμε να απορρίψουμε τον θόρυβο με τη χρήση του αλγορίθμου μας.

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> - Ο k-SVD ΑΛΓΟΡΙΘΜΟΣ ΚΑΙ Η ΑΡΑΙΗ ΑΝΑΠΑΡΑΣΤΑΣΗ

### 3.1 Εκμάθηση λεξικού

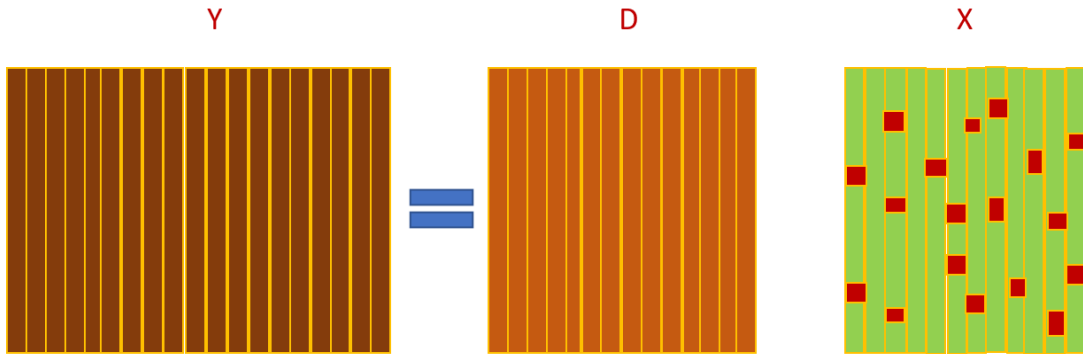
Για να αναπαραστήσουμε ένα δεδομένο σήμα σε αραιή κωδικοποίηση είναι απαραίτητο να υπολογίσουμε τους συντελεστές ενός αραιού διανύσματος με χρήση δεδομένου διανύσματος που συσχετίζεται με το αρχικό μας σήμα. Η μαθηματική σχέση που συνδέει αυτά τα δύο διανύσματα θεωρώντας ως  $y$  το αρχικό σήμα και ως  $x$  το διάνυσμα της αραιής κωδικοποίησης είναι  $y=Dx$ , όπου  $D$  είναι ένας πίνακας τον οποίο καλούμε υπερπλήρες λεξικό. Το πρόβλημα αυτό απεικονίζεται ακόλουθο Σχήμα 3.



Σχήμα 3: Το πρόβλημα Compressed Sensing (CS)

Συνήθως όταν έχουμε ένα σταθερό λεξικό (fixed dictionary) μας δημιουργείται πρόβλημα κατά την αναπαράσταση των σημάτων αφού δεν καθίσταται δυνατό να έχουμε ένα λεξικό που να είναι ικανό να καλύψει ένα σύνολο από παρόμοια σήματα. Έτσι δημιουργείται η ανάγκη να κατασκευάσουμε λεξικά προερχόμενα από δεδομένα του κάθε προβλήματος ατομικά δηλαδή από το αρχικό σήμα του. Αν δεν χρησιμοποιούμε σταθερό λεξικό προσθέτουμε έναν άγνωστο ακόμα στο αραιής κωδικοποίησης πρόβλημα. Το πρόβλημα αυτό είναι γνωστό και ως εκμάθηση λεξικού (Dictionary Learning –DL), όπως φαίνεται και στο Σχήμα 4, και συμπεριλαμβάνεται

στο να βρεθεί κατάλληλο λεξικό  $D$  και ένα αραιό διάνυσμα  $x$  έτσι ώστε να επιτύχουμε το μέγιστο της αραιής αναπαράστασης για το σήμα  $y$ .



Σχήμα 4: Το πρόβλημα Dictionary Learning (DL)

Ειδικότερα, ας υποθέσουμε ότι έχουμε  $N$  - διάστατα διανύσματα  $y_1, y_2, \dots, y_n$  που αποτελούν το σύνολο των αρχικών μας σημάτων. Αυτό που ζητάμε είναι να βρούμε ένα μητρώο - λεξικό  $D$  με διάσταση  $N \times \ell$  και με  $k$  αραιά  $\ell$  - διάστατα διανύσματα  $\theta_1, \theta_2, \dots, \theta_n$ , ώστε να ισχύει

$$y_i \approx D\theta_i \quad i=1,2,\dots,n \quad (27)$$

Αν χρησιμοποιήσουμε για το σφάλμα μας ως μετρική ποσοτικοποίηση την στάθμη Frobenius, το DL θα μπορεί να εκφραστεί και ως πρόβλημα βελτιστοποίησης σύμφωνα με τον τύπο (28)

$$\begin{aligned} \min_{D, \theta_1, \theta_2, \dots, \theta_n} \sum_{i=1}^n \|y_i - D\theta_i\|_F^2 \\ \text{s.t. } \|\theta_i\|_0 \leq k, i=1,2,\dots,n. \end{aligned} \quad (28)$$

Κάθε μία από της στήλες  $d_i$  προερχόμενες από το μητρώο - λεξικό  $D$  αντιστοιχεί σε ένα άτομο του λεξικού. Άρα το  $D$  λεξικό απαρτίζεται από  $\ell$  άτομα που το κάθε ένα έχει διάσταση  $N$ . Επιπλέον, για κάθε στήλη  $d_i$  του μητρώου (λεξικού)  $D$  κανονικοποιείται στη μοναδιαία Ευκλείδεια στάθμη, ήτοι  $\|d_i\|_2$ , αυτό συμβαίνει γιατί έτσι το λεξικό  $D$  και τα αραιά διανύσματα  $\theta_i$ , θα μπορούν να εξάγονται μοναδικά διανύσματα από έναν αλγόριθμο DL. Γενικά σε ένα υπερπλήρες λεξικό ισχύει  $\ell > N$  όταν το πλήθος των συντελεστών που είναι διάφοροι του μηδενός των αραιών διανυσμάτων  $\theta_i$  είναι αρκετά μικρότερο της διάστασης των σημάτων, ήτοι  $k \ll N$ .

Όταν για τα διανύσματα  $\theta_i$  έχουμε βαθμό αραιότητας  $k=1$ , το DL πρόβλημα μετατρέπεται σε πρόβλημα συσταδοποίησης (clustering). Σε μία τέτοια περίπτωση ο αλγόριθμος DL λειτουργεί με σκοπό τη δημιουργία κατάλληλου λεξικού  $D$  έτσι ώστε να είναι ευκολότερη η εύρεση του καταλληλότερου ατόμου του λεξικού που θα αναπαραστήσει κάθε σήμα  $y_i$  ως γινόμενο από ένα συντελεστή και ένα άτομο.

Προκειμένου να επιτύχουμε την ποσοτικοποίηση για το σφάλμα που προκύπτει στην εξίσωση (28) έχουμε τη δυνατότητα να χρησιμοποιήσουμε επιπλέον μετρικές από αυτές που είχαμε έως τώρα, παραδείγματος χάρη τη στάθμη  $\ell_1$ .

Η γραφή του προβλήματος (28) μπορεί να γίνει και με τη βοήθεια μητρώων αφού θεωρήσουμε ως  $Y = [y_1, y_2, \dots, y_n]$  το μητρώο σημάτων, και ως  $\Theta = [\theta_1, \theta_2, \dots, \theta_n]$  το μητρώο των συντελεστών των αραιών διανυσμάτων. Έτσι, το πρόβλημα γράφεται ως εξής:

$$\begin{aligned} \min_{A, Z} \|X - AZ\|_F^2 \\ \text{s.t. } \|y_i - D\theta_i\|_2^2 \leq \varepsilon, i=1,2,\dots,n. \end{aligned} \quad (29)$$

και μετατρέπεται σε ένα από τα προβλήματα παραγοντοποίησης μητρώων (matrix factorization problem).

Ακόμα, υπάρχει περίπτωση στο πρόβλημα βελτιστοποίησης (28) να έχουμε ως σκοπό να επιτύχουμε την δυνατότερη αραιότερη αναπαράσταση των σημάτων μας με δεδομένο ένα κατώφλι σφάλματος  $\varepsilon$ .

Σε αυτή την περίπτωση το πρόβλημά μας θα γράφονταν ως

$$\begin{aligned} \min_{D, \theta_1, \theta_2, \dots, \theta_n} \sum_{i=1}^n \|y_i - D\theta_i\|_F^2 \\ \text{s.t. } \|\theta_i\|_0 \leq k, i=1,2,\dots,n \end{aligned} \quad (30)$$

Στις δύο περιπτώσεις του προβλήματος DL, όπως γράφονται στις εξισώσεις (28) και (30), δύναται να συγχωνευθούν σε μία, αν αντί για τα κατώφλια αραιότητας  $k$  και με σφάλμα  $\varepsilon$ , χρησιμοποιηθεί μία παράμετρος κανονικοποίησης  $\lambda$  έτσι ώστε να εκφράζει το βάρος που θέλουμε να έχει η αραιότητα και το σφάλμα προσέγγισης. Με μία τέτοια παράμετρο ένα πρόβλημα DL γράφεται ως

$$\min_{D, \theta_1, \theta_2, \dots, \theta_n} \sum_{i=1}^n \|y_i - D\theta_i\|_F^2 + \lambda \|\theta_i\|_0 \quad (31)$$

Όσο μεγαλύτερου μεγέθους παράμετρο κανονικοποίησης  $\lambda$  έχουμε, τόσο αυξάνεται και η βαρύτητα που δίνουμε στην αραιότητα του διανύσματος μέσω της στάθμης  $\ell_0$  και συνεπώς η αναπαράσταση στο σήμα είναι όσο γίνεται πιο αραιή σε βάρος του προσεγγιστικού σφάλματος  $\epsilon$ . Αντιθέτως, όσο πιο μικρή είναι η παράμετρος κανονικοποίησης  $\lambda$ , τόσο μικρότερο είναι το προσεγγιστικό σφάλμα σε βάρος της αραιότητας.

Η στάθμη  $\ell_0$  στα προβλήματα DL συμβάλλει στην ανάγκη να αναπτυχθούν μέθοδοι εξαντλητικής αναζήτησης με σκοπό να βρεθεί λύση στο πρόβλημα βελτιστοποίησης (πρόβλημα κλάσης NP). Σε πολλές περιπτώσεις τη θέση της στάθμης  $\ell_0$  παίρνει η στάθμη  $\ell_1$ , αφού όπως έχει επισημανθεί προηγουμένως η  $\ell_1$  στάθμη είναι αυτή η οποία είναι πιο κοντά στην  $\ell_0$  η οποία συντηρεί το κυρτό του χαρακτήρα του προβλήματος. Σε αυτή την περίπτωση το πρόβλημα βελτιστοποίησης (31) γράφεται ως

$$\min_{D, \theta_1, \theta_2, \dots, \theta_n} \sum_{i=1}^n \|y_i - D\theta_i\|_F^2 + \lambda \|\theta_i\|_1 \quad (32)$$

και είναι το κυρτό πρόβλημα του οποίου η λύση προσεγγίζει την καλύτερη λύση του προβλήματος DL.

Παρόλο που τόσο το πρόβλημα CS όσο αντίστοιχα και το πρόβλημα DL συμπεριλαμβάνεται στο παραπάνω πρόβλημα βελτιστοποίησης με ένα ή και δύο αγνώστους, ο τελικός στόχος τους είναι διαφορετικός. Το πρόβλημα CS έχει ως στόχο, αφού επιλέξουμε προσεκτικά το σταθερό λεξικό  $D$ , να βρούμε τα καταλληλότερα αραιά διανύσματα, για να αναπαραστήσουμε το αρχικό σήμα. Αντίθετα στο πρόβλημα DL, χρησιμοποιούμε το αρχικό μας σήμα ως δεδομένο εκπαίδευσης, με σκοπό να επιλέξουμε τα καταλληλότερα άτομα για να δημιουργήσουμε το λεξικό  $D$ . Ακόμα, ο υπολογισμός του αραιού διανύσματος από τον αλγόριθμο DL, έχει ως στόχο τον εντοπισμό και την απεικόνιση της πιο βέλτιστης γραμμικής αναπαράστασης του σήματος από το λεξικό.

### 3.2. Γενική περιγραφή του αλγορίθμου (k-SVD)

Ο K-SVD είναι ένας αλγόριθμος που χρησιμοποιείται για την αναπαράσταση σήματος. Συγκεκριμένα, μέσα από ένα σύνολο σημάτων μπορεί να εξαγάγει ένα λεξικό ικανό να προσεγγίσει κάθε σήμα με έναν αραιό συνδυασμό ατόμων. Οι Marc Lebrun and Arthur Leclaire το 2012 εστίασαν στην δημιουργία αλγορίθμου διαγραφής θορύβου εικόνας που βασίζεται σε K-SVD [9]. Οι Aharon, Elad και Bruckstein το 2006 πρότειναν τον αλγόριθμο k-SVD [1] που είναι ιδιαίτερα γνωστός για την επίλυση προβλημάτων επεξεργασίας σήματος όπως για παράδειγμα η αποθορυβοποίηση και η συμπίεση σημάτων. Ο σκοπός ενός τέτοιου αλγορίθμου είναι να αναπαραστήσει ένα σήμα σαν γινόμενο αραιού διανύσματος και ενός υπερπλήρους λεξικού όσο γίνεται με τον πιο ακριβή τρόπο. Σε σχέση με τους αλγόριθμους LASSO και OMP η διαφορά του είναι στο μη σταθερό υπερπλήρες λεξικό του αλλά και στο ότι βήμα με βήμα μεταβάλετε όπως και το αραιό διάνυσμα συντελεστών.

Στον αλγόριθμο εκμάθησης λεξικού K-SVD υπάρχουν διαφορετικές μέθοδοι για την αφαίρεση διαφορετικών επιπέδων θορύβων που υπάρχουν σε μία εικόνα. Με τον αλγόριθμο K-SVD μπορούμε να αφαιρέσουμε το θόρυβο από έγχρωμες, ασπρόμαυρες και γκρι εικόνες [9]. Ο k-SVD αλγόριθμος προσπαθεί στην ουσία βήμα βήμα να βρει το διάνυσμα που είναι πιο αραιό και το πιο καλό υπερπλήρες λεξικό για να αναπαραστήσει το σήμα. Ειδικότερα, έστω ότι υπάρχουν  $N$  δεδομένα και  $l$ -διάστατα σήματα  $x_n, n=1,2,\dots,n$ , τα οποία και θέλουμε να αναπαραστήσουμε το κάθε ένα με την μορφή γινομένου ως εξής:

$$x_n = Az_n \quad (33)$$

όπου  $A$  είναι το μητρώο που παριστάνει το υπερπλήρες λεξικό διάστασης  $l \times m$  το οποίο είναι το αραιό διάνυσμα συντελεστή. Ο πραγματικός στόχος είναι να έχουμε όσο γίνεται καλύτερη προσέγγιση του σήματος με τον περιορισμό του αραιού συντελεστή διανύσματος και όχι την ακριβή αναπαράστασή του. Έτσι το πρόβλημα βελτιστοποίησης μπορεί να γραφτεί ως:

$$\begin{aligned} & \min_{A,Z} \|X - AZ\|_F^2 \\ & \text{s.t.} \quad \|z_n\|_0 \leq T_0, n=1,2,\dots,N \end{aligned} \quad (34)$$

όπου  $X = [x_1, x_2, \dots, x_n]$  είναι το μητρώο, με  $I \times N$  που έχει ως στήλες τα διανύσματα που αναπαριστούν τα σήματα  $x_n$ , το μητρώο  $Z = [z_1, z_2, \dots, z_n]$  είναι το  $m \times N$  μητρώο με στήλες τα αραιά διανύσματα, και  $T_0$  το κατώφλι που προσδιορίζει το βαθμό αραιότητας των διανυσμάτων.

Έχοντας ως στόχο την επίλυση του προβλήματος ελαχιστοποίησης, ένας κλασικός τρόπος αντιμετώπισής του είναι να διαιρούμε κάθε βήμα του αλγορίθμου σε δύο στάδια τα οποία είναι ανεξάρτητα μεταξύ τους. Στο πρώτο στάδιο που ονομάζουμε και αραιή κωδικοποίηση (sparse coding) ο αλγόριθμός μας κρατάει σταθερό το υπερπλήρες λεξικό  $A$  και με τη βοήθεια ενός άλλου αλγορίθμου σαν τον OMP ή το LASSO βρίσκει το καλύτερο αραιό διάνυσμα με συντελεστή  $z_n$  για κάθε σήμα  $x_n$ . Στο επόμενο στάδιο, που το ονομάζουμε ενημέρωση λεξικού (codebook update) κρατάμε σταθερούς τους συντελεστές  $z_n$ , και ψάχνουμε την καλύτερη λύση για τις στήλες του λεξικού  $A$ . Τα δύο αυτά στάδια του αλγορίθμου περιγράφονται αναλυτικότερα ως εξής:

1. Αραιή κωδικοποίηση. Έστω ότι θεωρούμε σταθερό το λεξικό  $A$ , όπως έχει δημιουργηθεί από την τελευταία επανάληψη που εκτέλεσε ο αλγόριθμος και επιλύει το εξής πρόβλημα βελτιστοποίησης:

$$\begin{aligned} & \min_Z \|X - AZ\|_F^2 \\ & \text{s.t.} \quad \|z_n\|_0 \leq T_0, n=1,2,\dots,N. \end{aligned} \quad (35)$$

Το οποίο ισοδυναμεί με την επίλυση  $N$  προβλημάτων βελτιστοποίησης της μορφής

$$\begin{aligned} & \min_{z_n} \|x_n - Az_n\|_F^2 \\ & \text{s.t.} \quad \|z_n\|_0 \leq T_0, n=1,2,\dots,N. \end{aligned} \quad (36)$$

Αντίστοιχα, αν ο περιορισμός δεν αντιστοιχεί σε κατώφλι αραιότητας  $T_0$  αλλά είναι το προσεγγιστικό σφάλμα  $\varepsilon$  στη στάθμη της διαφοράς του σήματος  $x_n$ , από την εκτίμησή του  $Az_n$ , τότε το προαναφερόμενο πρόβλημα βελτιστοποίησης μπορεί να γραφεί ως

$$\begin{aligned} \min_{z_n} \|z_n\|_0 \\ \text{s.t. } \|x_n - Az_n\|^2 < \varepsilon, n=1,2,\dots,N. \end{aligned} \quad (37)$$

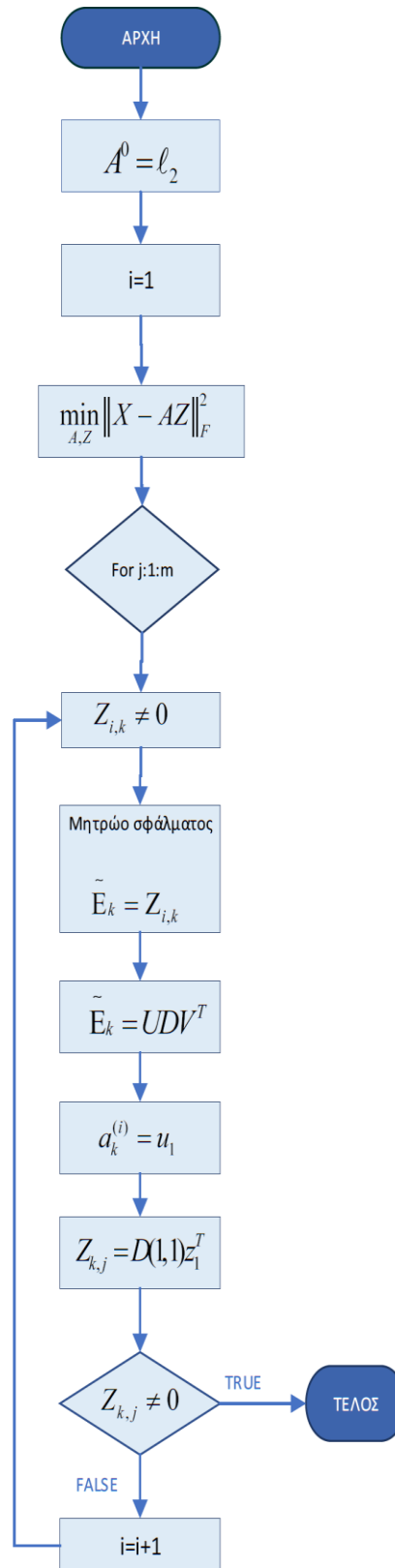
2. Ενημέρωση λεξικού. Έχοντας ως δεδομένο ότι στο αρχικό στάδιο του αλγορίθμου έχουμε πάρει τα καλύτερα δυνατά αραιά διανύσματα συντελεστή  $z_n$ , ο στόχος στο δεύτερο στάδιο είναι να βελτιστοποιήσουμε κάθε στήλη του λεξικού  $A$ . Έστω ότι είμαστε στο βήμα ενημέρωσης της στήλης  $a_k$  του μητρώου  $A$  με σκοπό να ελαχιστοποιήσουμε την ποσότητα  $\|X - AZ\|_F^2$ . Το γινόμενο των μητρώων  $AZ$  μπορεί να γραφεί ως το άθροισμα των μητρώων με βαθμό ένα ως εξής:

$$AZ = \sum_{i=1}^m a_i z_i^T \quad (38)$$

Όπου τα  $z_i^T$ ,  $i=1,2,\dots,m$  είναι γραμμές του μητρώου  $Z$ .

Ο αλγόριθμος σε μορφή διαγράμματος ροής παρουσιάζεται στο Σχήμα 5.





Σχήμα 5: Ο αλγόριθμος k-SVD

Ο  $k$ -SVD αλγόριθμος μας προσφέρει αρκετά πλεονεκτήματα αφού χαρακτηρίζεται από απλότητα και ταχύτητα κατά την εφαρμογή του εξάγοντας τις περισσότερες φορές το βέλτιστο αποτέλεσμα. Η ταύτιση του αλγορίθμου  $k$ -SVD με την βέλτιστη λύση εξαρτάται από το πόσο αποτελεσματικός είναι ο αλγόριθμος που χρησιμοποιείται στο βήμα της κωδικοποίησης αραιής αναπαράστασης.

Η διαδικασία που περιγράφεται από τον  $k$ -SVD αλγόριθμο είναι στην ουσία η γενίκευση του αλγορίθμου  $k$ -means που είναι ο κλασικός αλγόριθμος που χρησιμοποιούμε για την επίλυση προβλημάτων συσταδοποίησης (clustering). Συγκεκριμένα, ο αλγόριθμος  $k$ -means αποτελεί μία υποπερίπτωση  $k$ -SVD αλγορίθμου με κατώφλι αραιότητας  $T_0$  ίσο με τη μονάδα.

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> - ΑΠΟΘΟΥΒΟΠΟΙΗΣΗ ΕΙΚΟΝΑΣ ΜΕ ΧΡΗΣΗ k-SVD

### 4.1 Γενικά

Στο κεφάλαιο περιγράφεται το πώς εφαρμόζεται τον αλγόριθμο k-SVD με σκοπό την αποθρομβοποίηση μιας εικόνας. Για να μπορέσουμε να το οπτικοποιήσουμε χρησιμοποιούμε εικόνες grayscale με αναπαράσταση χρωμάτων 8 bits. Επιδρούμε στις εικόνες μας με διαφορετικών ειδών θόρυβο προκειμένου να δούμε την αποτελεσματικότητα του k-SVD αλγόριθμου που εφαρμόζουμε. Για κάθε έναν από τον θόρυβο που εφαρμόζουμε και μελετάμε βλέπουμε τρεις εικόνες, την αρχική μας εικόνα, την εικόνα με θόρυβο και τέλος την αποθρομβοποιημένη μετά την εφαρμογή του αλγόριθμου. Για να ποσοτικοποιήσουμε τα αποτελέσματα σε κάθε εικόνα χρησιμοποιούμε τον λόγο Peak Signal-to-Noise Ratio (PSNR) σε dB η οποία ορίζεται ως εξής:

$$PSNR_{dB} = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \quad (39)$$

όπου η ποσότητα MSE αντιστοιχεί στο μέσο τετραγωνικό σφάλμα (Mean Squared Error) του σήματος που προκύπτει από την εικόνα σε αντιστοιχία με το αρχικό σήμα της. Ειδικότερα αν I η αρχική μας εικόνα και K η εικόνα που είναι η προσέγγιση της αρχικής, τότε το MSE δίνεται από τον τύπο:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (40)$$

Επιπλέον, το  $MAX_I$  που συναντήσαμε στην εξίσωση (39) είναι η μέγιστη τιμή που μπορεί να έχει η ένταση του pixel της εικόνας I. Άρα στην περίπτωση μας, οι 8-bits εικόνες αντιστοιχούν με  $MAX_I = 256$ .

### 4.2. Μέθοδος των επικαλυπτόμενων τμημάτων

Χρησιμοποιούμε την τεχνική των επικαλυπτόμενων περιοχών (overlapped patches) για να εφαρμόσουμε έναν αλγόριθμο DL ώστε να αποθρομβοποιήσουμε μια εικόνα όπως περιγράφεται στην εργασία [6]. Σύμφωνα με την μέθοδο αυτή έχοντας

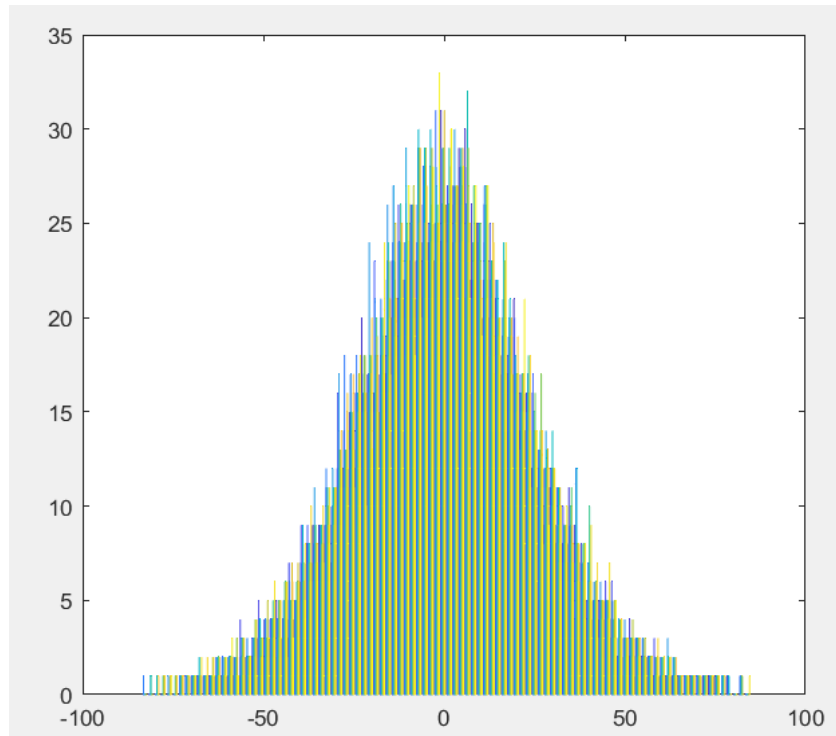
αρχικά μία εικόνα διάστασης  $n \times n$  μπορούμε να εξάγουμε όλες τις περιοχές αυτής της εικόνας με μέγεθος αυτής της υποπεριοχής  $m \times m$  όπου  $m$  πολύ μικρότερο του  $n$ . Άρα εύκολα υπολογίζουμε ότι μία εικόνα διάστασης  $n \times n$  αποτελείται από  $(n - m + 1)^2$  επικαλυπτόμενα τμήματα με το καθένα να έχει μέγεθος  $m \times m$ , ενδεικτικά αν έχουμε μία εικόνα που περιγράφεται από τετραγωνικό μητρώο μεγέθους 256 από αυτή εξάγονται  $(256 - 12 + 1)^2 = 60025$  επικαλυπτόμενα τμήματα, για τετραγωνικό μητρώο  $12 \times 12$ . Κατόπιν μετατρέπουμε κάθε τμήμα σε διάνυσμα (στήλης) το οποίο έχει πλήθος στοιχείων  $m^2$ . Εάν γίνει παράθεση αυτών των διανυσμάτων το ένα δίπλα στο άλλο προκύπτει ένα μητρώο με διάσταση  $m^2 \times (n - m + 1)^2$  που είναι το μητρώο του αρχικού σήματος συμπεριλαμβανομένου του θορύβου. Για την ανασύσταση του σήματος της εικόνας για κάθε pixel υπολογίζουμε το μέσο όρο ενός τετραγωνικού μητρώου διάστασης  $n$ . Μία μέθοδος κατασκευής είναι η δημιουργία μητρώου έστω  $W$  (όπου η διάσταση  $W$  είναι η διάσταση της εικόνας) στο οποίο το εικονοστοιχείο που βρίσκεται στην  $i$  γραμμή και  $j$  στήλη απεικονίζει το πλήθος των τμημάτων στα οποία συμμετέχει κάθε εικονοστοιχείο. Αν θεωρήσουμε μία εικόνα μεγέθους  $8 \times 8$  και ένα τμήμα τετραγωνικής διάστασης 2 το μητρώο  $W$  απεικονίζεται ως ακολούθως:

$$W = \begin{pmatrix} 1 & 2 & \dots & 2 & 1 \\ 2 & 4 & \dots & 4 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 4 & \dots & 4 & 2 \\ 1 & 2 & \dots & 2 & 1 \end{pmatrix} \quad (41)$$

### 4.3 Αποθορυβοποίηση σήματος με θόρυβο Gaussian

Θεωρούμε ότι προσθετικός θόρυβος επιδρά σε μία εικόνα, ο οποίος ακολουθεί κανονική κατανομή [9] [21]. Ο Gaussian θόρυβος είναι μέσης τιμής ίσης με μηδέν στα μοντέλα που χρησιμοποιούμε δηλαδή η τιμή της πυκνότητας πιθανότητας θα είναι όπως φαίνεται ακολούθως:

$$p_N(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{z^2}{2\sigma^2}} \quad (42)$$



Σχήμα 6. Κατανομή gauss για  $\sigma=20$

Η κατανομή Gauss θορύβου διασποράς ίση με 20, η οποία παρουσιάζεται στο Σχήμα 6 έχει επιπρόσθετη επίδραση σε αρχείο εικόνας διαστάσεων 512x512 εικονοστοιχείων (σημ: κάθε εικονοστοιχείο αναπαρίσταται με χρήση 8 bits άρα ανήκει στο διάστημα  $[0,255]$ ). Αυτό που θέλουμε να επιτύχουμε είναι να ανιχνεύσουμε και να εξαλείψουμε αυτού του είδους το θόρυβο έχοντας γνώση το σήμα που αναπαριστά η εικόνα με ενσωματωμένο το θόρυβο.

Ακολουθούμε τη διαδικασία που περιγράψαμε στην προηγούμενη ενότητα για την εφαρμογή του αλγορίθμου k-SVD στην εικόνα που περιέχει το θόρυβο. Στην περίπτωση μας η αρχική εικόνα είναι ένα διάνυσμα 512x512 το οποίο μετατρέπουμε με χρήση των επικαλυπτόμενων τμημάτων (overlapped patches) διάστασης 8x8 σε διάνυσμα διάστασης 64x255025. Για κάθε μια από τις στήλες του μητρώου μας αναπαριστάμε ένα τμήμα εικόνας με τα στοιχεία του διάστασης 8x8 να έχουν υποστεί αναδιάταξη ως προς τις στήλες, έτσι δημιουργείται ένα διάνυσμα στήλης αποτελούμενο από 64 στοιχεία. Έπειτα εφαρμόζουμε τον αλγόριθμο για την

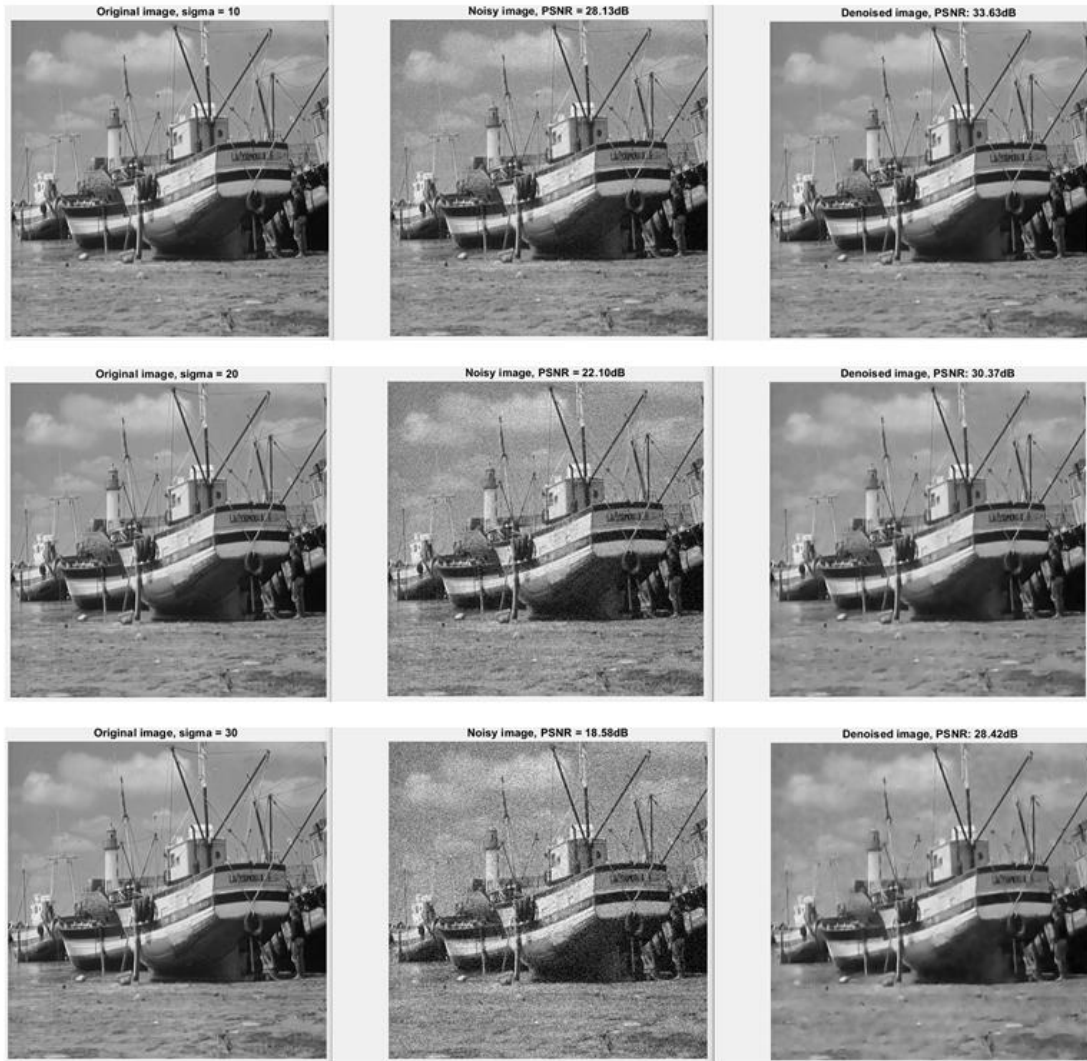
αναπαράσταση κάθε τμήματος ως γινόμενο ενός (αραιού) διανύσματος επί ενός λεξικού το οποίο εξάγουμε από το αρχικό εν θόρυβο σήμα. Για αρχικό λεξικό χρησιμοποιείται ένα υπερπλήρες λεξικό του χώρου δύο διαστάσεων μετασχηματισμού DCT με τον τρόπο που έχει περιγράψει προηγουμένως. Ο αλγόριθμος μας δίνει ως έξοδο ένα μητρώο  $64 \times 256$  το οποίο στην ουσία είναι το λεξικό μας, αποτελούμενο από 256 τμήματα και ένα μητρώο  $256 \times 256 \times 0.25$  στο οποίο έχουμε αναπαράσταση κάθε διανύσματος από στήλη. Αν πολλαπλασιάσουμε το λεξικό με κάθε αραιό διάνυσμα θα εξάγουμε νέα διανύσματα τα οποία αν αναδιαταχθούν τα στοιχεία τους μετατρέπονται σε τμήματα διάστασης  $8 \times 8$ . Τέλος, με βοήθεια ένα μητρώο βαρών όμοιου με αυτό της εξίσωσης (42) εξάγουμε την τιμή της έντασης για κάθε pixel της καινούργιας εικόνας ως μέσο όρο των τιμών του pixel στα διαφορετικά patches στα οποία συμμετέχει.



Εικόνα 4.1.a Image\_1 με block size 8, dict size 256 & sigma 10,20,30

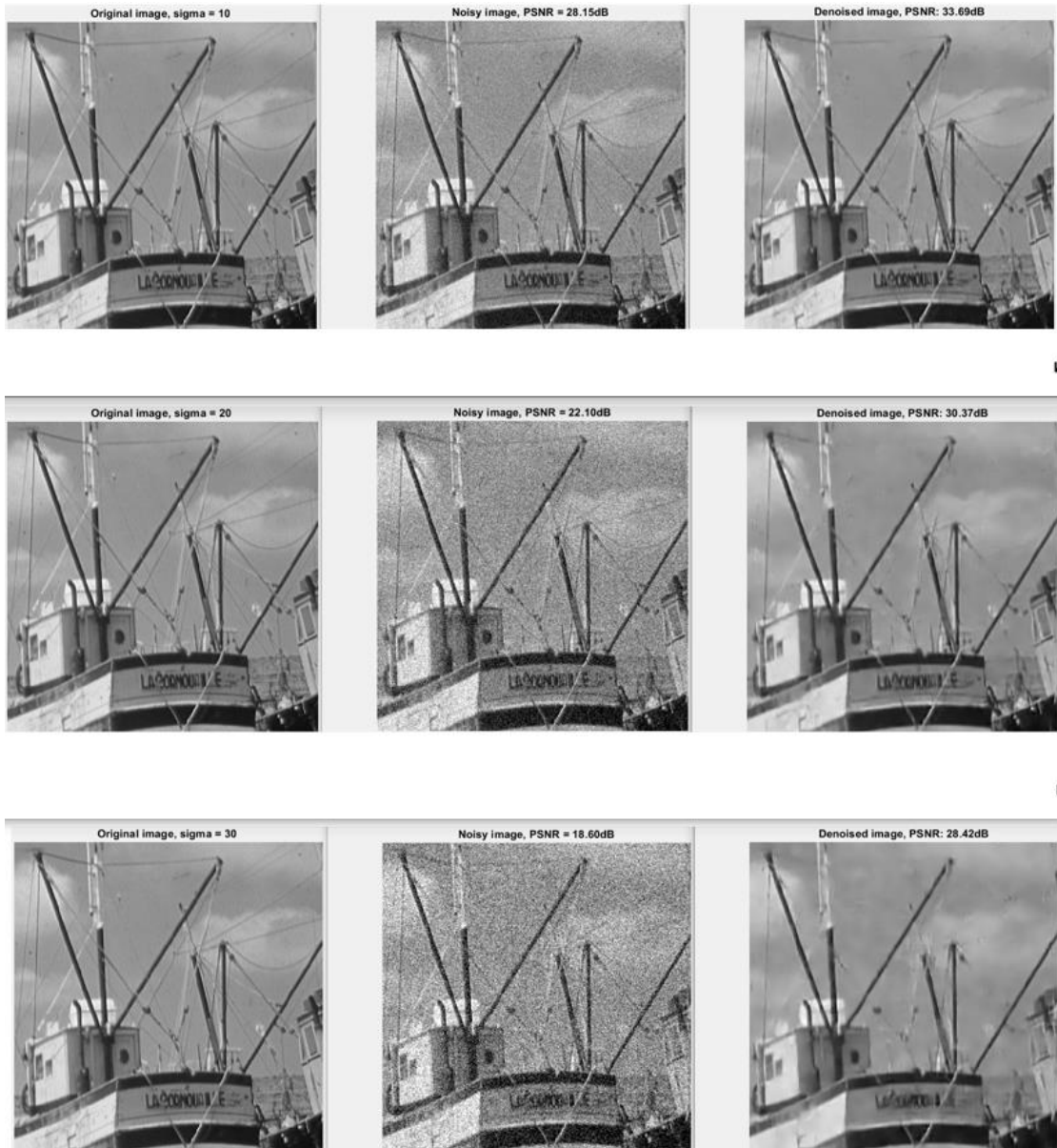


Εικόνα 4.1.b Image\_1 με block size 8, dict size 256 & sigma 10,20,30

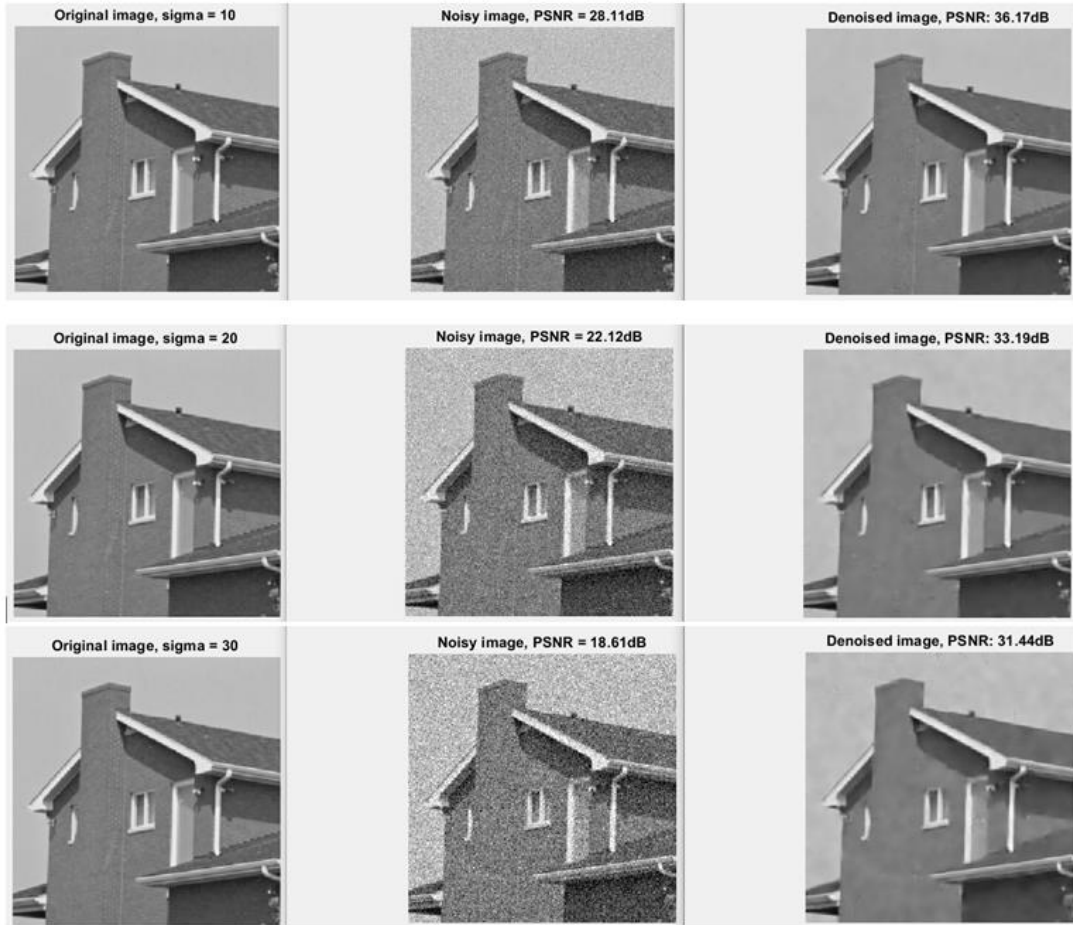


Εικόνα 4.2.α Image\_2 με block size 8, dict size 265 & sigma 10,20,30





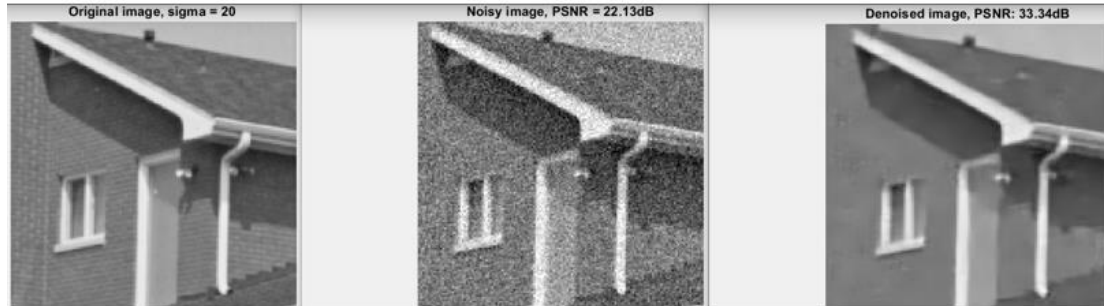
Εικόνα 4.2.b Image\_2 με block size 8, dict size 265 & sigma 10,20,30



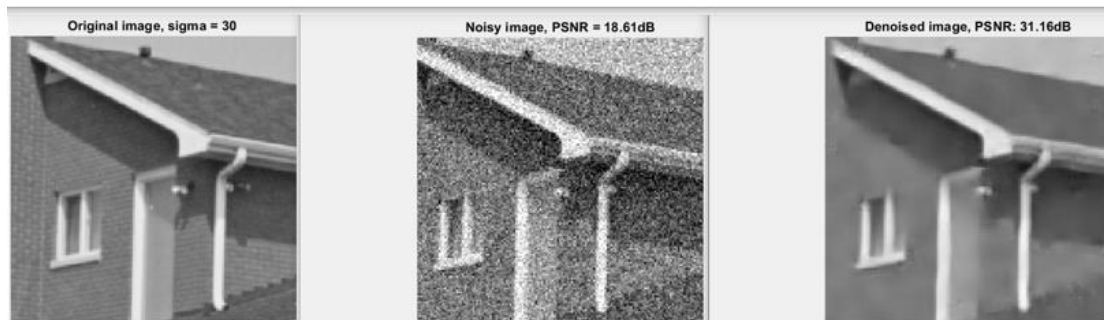
Εικόνα 4.3.α Image\_3 με block size 8, dict size 256 & sigma 10,20,30



κ



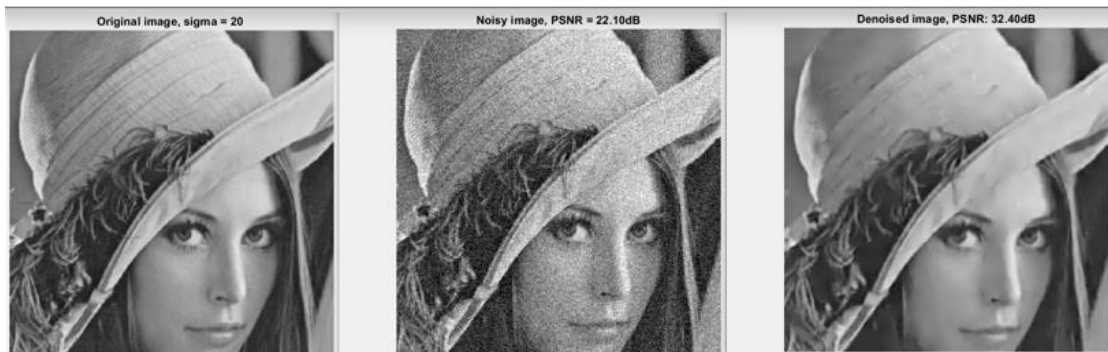
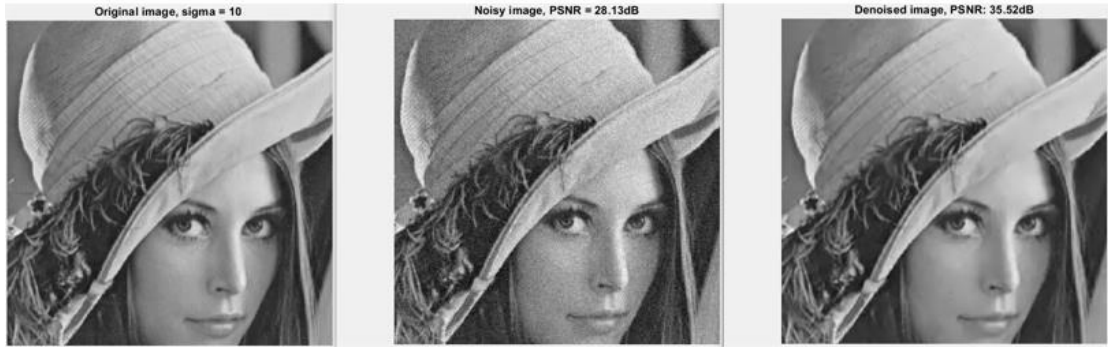
λ



Εικόνα 4.3.b Image\_3 με block size 8, dict size 256 & sigma 10,20,30



Εικόνα 4.4.α Image\_4 με block size 8, dict size 256 & sigma 10,20,30



Εικόνα 4.4.b Image\_4 με block size 8, dict size 256 & sigma 10,20,30

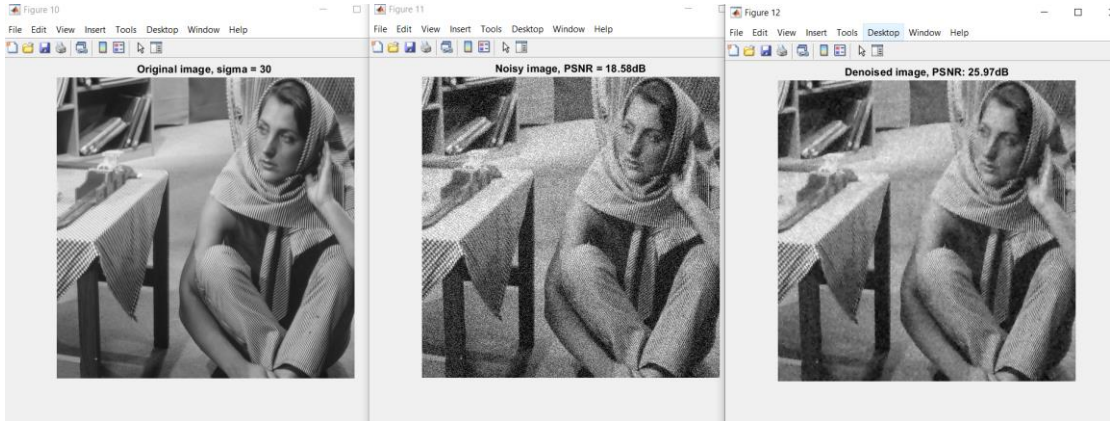


Εικόνα 4.5.a Image\_5 με block size 8, dict size 256 & sigma 10,20,30

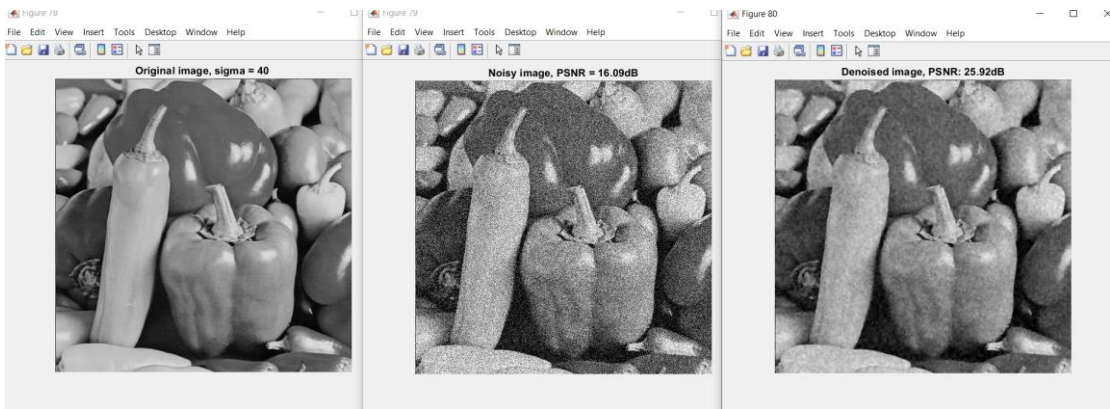


Εικόνα 4.5.b Image\_5 με block size 8, dict size 256 & sigma 10,20,30

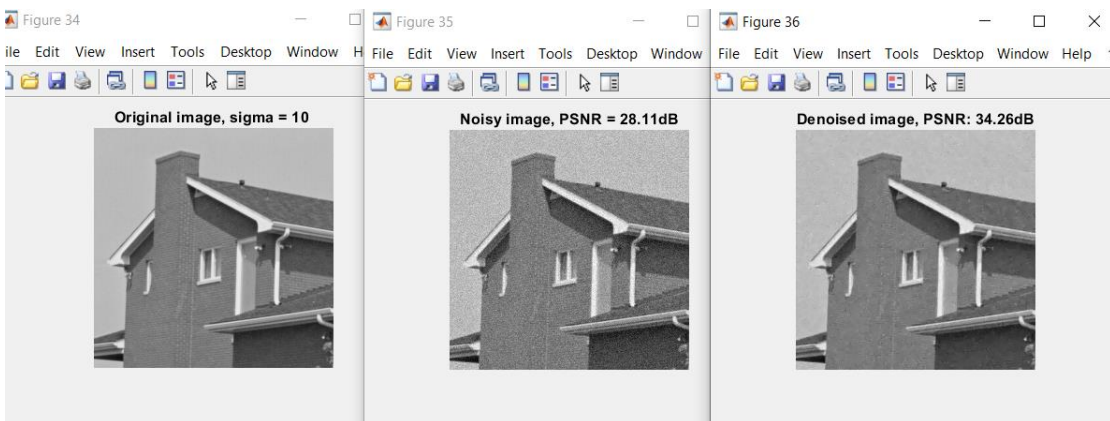
Τεχνικές αραίες κωδικοποίησης και οι εφαρμογές τους στην αποθρομβοποίηση και αποκατάσταση ψηφιακών εικόνων



Εικόνα 4.6 Image\_1, με Θόρυβος Gauss blocksize 8 & sigma 30



Εικόνα 4.7 Image 5 με Θόρυβο Gauss blocksize 4 & sigma 40



Εικόνα 4.8 Image\_3, με Θόρυβο Gauss blocksize 4 & sigma 10



Ένα από τα πιο κρίσιμα σημεία είναι να επιλέξουμε την κατάλληλη παράμετρο αραιότητας του αλγορίθμου k-SVD με σκοπό να γίνει όσο γίνεται μεγαλύτερη η απόρριψη θορύβου. Στην περίπτωση όπου έχουμε Gaussian θόρυβο πειραματικά αποτελέσματα [16] [21] έχουν δείξει ως καλύτερη επιλογή για το κατώφλι τιμή για το σφάλμα ίση με:

$$\varepsilon = 1.15 * m \times \sigma \quad (43)$$

Ο πρώτος παράγοντας είναι η διάσταση πλαισίου ενώ ο δεύτερος η παράμετρος της διασποράς στην κατανομή Gauss. Η συγκεκριμένη τιμή εντοπίζεται και στον OMP αλγόριθμο και αναφέρεται ως ακτίνα. Την ακτίνα αυτή χρησιμοποιούμε στο βήμα της αραιής κωδικοποίησης( περίπτωση k-SVD).

Η τιμή αυτή που παίρνει το κατώφλι σφάλματος δημιουργεί αρκετά μικρές τιμές στο κατώφλι της αραιής κωδικοποίησης, δηλαδή έχουμε τιμές να τείνουν στο ένα. Η τιμή του κατωφλίου της αραιής κωδικοποίησης δεν είναι εξαρτώμενη αποκλειστικά από το κατώφλι σφάλματος αλλά εξαρτάται και από το μέγεθος του τμήματος και τον αριθμό των ατόμων που περιέχονται στο λεξικό. Θεωρώντας τετραγωνικό τμήμα διάστασης  $m=8$  και αριθμό από άτομα στο λεξικό  $K=256$ , ο αλγόριθμος k-SVD για την αποθορυβοποίηση μιας εικόνας  $512 \times 512$  με Gaussian θόρυβο διασποράς 20 έχει ως αποτέλεσμα να παίρνει τιμή κατωφλίου αραιότητας  $T=1,57$ . Άρα για οποιοδήποτε αραιό διάνυσμα με διάσταση  $256 \times 1$  έχει περίπου 1.57 στοιχεία διάφορα του μηδενός.

Ο αλγόριθμος k-SVD αν εφαρμοστεί πάνω σε ένα σήμα που επηρεάζεται από θόρυβο Gaussian μας δίνει πολύ καλά αποτελέσματα. Βασική προϋπόθεση για να έχουμε το καλύτερο αποτέλεσμα στην ανάκτηση του αρχικού σήματος είναι η εκ των προτέρων γνώση της διασποράς του Gaussian θορύβου. Στο sparse coding step χρησιμοποιείται η παράμετρος της διασποράς του αλγορίθμου και μας δείχνει το μέτρο της αραιότητας που θέλουμε να πετύχουμε στην αναπαράσταση του σήματος με τη χρήση ενός υπερπλήρες λεξικού.

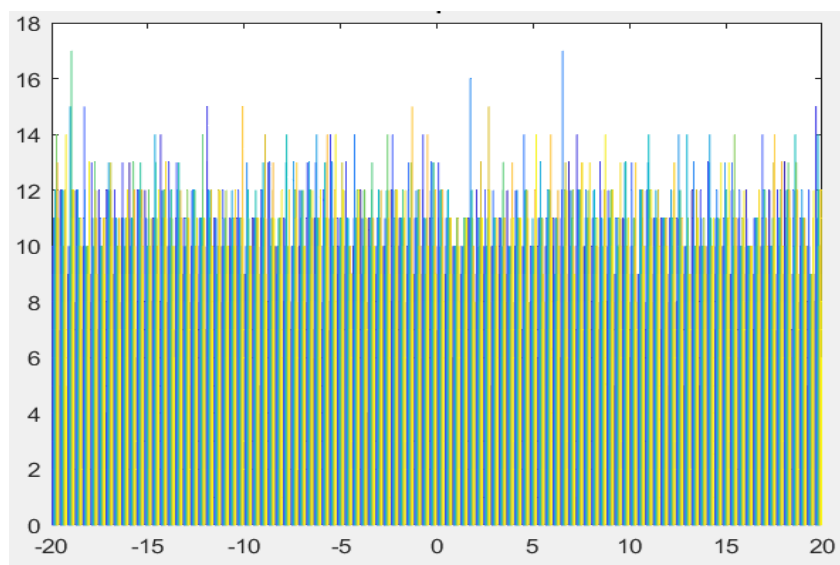
Για να είναι κατάλληλη η τιμή της παραμέτρου στο sparse coding step του αλγορίθμου, είναι απαραίτητο να γνωρίζουμε από πριν τη διασπορά του Gaussian θορύβου από τον οποίο έχει επηρεαστεί το αρχικό σήμα της εικόνας μας. Ο

αλγόριθμος k-SVD μας δίνει αρκετά καλά αποτελέσματα όταν απευθύνεται σε κανονικές κατανομές θορύβου. Αυτό μπορεί να χρησιμεύσει ώστε να εφαρμοστεί ο αλγόριθμος και σε διαφορετικές περιπτώσεις θορύβου όπως είναι ο Poisson θόρυβος ή ο θόρυβος με ομοιόμορφη κατανομή χωρίς απαραίτητως τα αποτελέσματα να είναι το ίδιο καλά, αλλά ικανοποιητικά σε ένα βαθμό.

#### 4.4. Αφαίρεση ομοιόμορφου θορύβου από εικόνα

Στην περίπτωση που το σήμα μας έχει θόρυβο προερχόμενο από ομοιόμορφη κατανομή τότε χρησιμοποιούμε τον ίδιο αλγόριθμο όπως στην περίπτωση που είχαμε θόρυβο Gaussian. Έτσι, αν ο θόρυβος ακολουθεί ομοιόμορφη κατανομή σε ένα διάστημα  $[a, b]$  με τύπο

$$f(x) = \frac{1}{b-a+1} \quad (44)$$



Σχήμα 7: Κατανομή με θόρυβο ομοιόμορφης κατανομής και  $\sigma=40$

τότε η διασπορά του θορύβου είναι  $\sigma = \frac{(b-a+1)^2 - 1}{12}$  η οποία χρησιμοποιείται

ως η παράμετρος στο sparse coding step του αλγορίθμου. Στο σχήμα (7) φαίνεται η εφαρμογή του αλγορίθμου πάνω σε μία εικόνα η οποία έχει επηρεαστεί από θόρυβο ομοιόμορφης κατανομής στα διαστήματα  $[-20,20]$ .

Ο αλγόριθμος k-SVD, όπως εφαρμόσαμε έως τώρα εκλαμβάνει το θόρυβο ομοιόμορφης κατανομής ως Gaussian θόρυβο με διασπορά που υπολογίζεται με βάση τις τιμές των pixels της εικόνας με τον θόρυβο. Η θεώρηση αυτή δίνει αρκετά καλά αποτελέσματα αν και η κατανομή του θορύβου είναι με βάση τα σχήματα (6) και (7) φαίνεται τελείως διαφορετική η κατανομή. Στην ομοιόμορφη κατανομή παρατηρούμε ότι όσο μικρότερο «sigma» έχουμε τόσο καλύτερα αποτελέσματα παίρνουμε στην αποθρομβοποίηση της εικόνας. Στις εικόνες 4.1 έως 4.8 που ακολουθούν φαίνεται η εφαρμογή του αλγορίθμου πάνω σε 5 διαφορετικές εικόνες που μελετήσαμε και με τιμές θορύβου για  $\sigma=10,20,30,40$  αντίστοιχα.



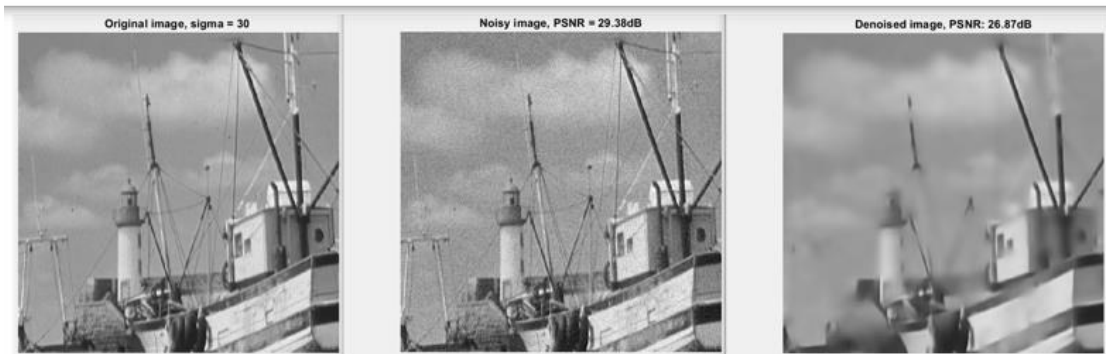
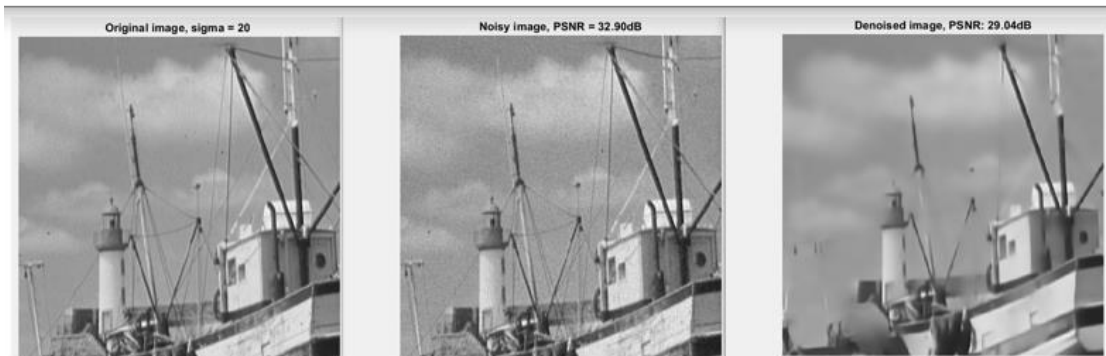
Εικόνα 4.9.a Image\_1 με block size 8, dict size 512 & sigma 10,20,30



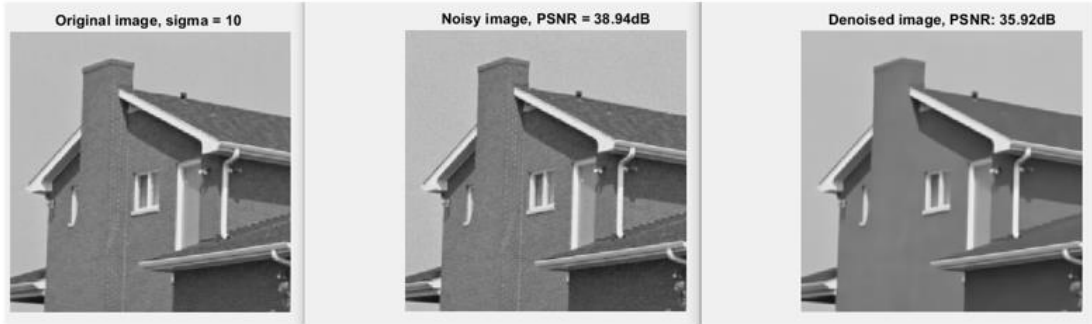
Εικόνα 4.9.b Image\_1 με block size 8, dict size 512 & sigma 10,20,30



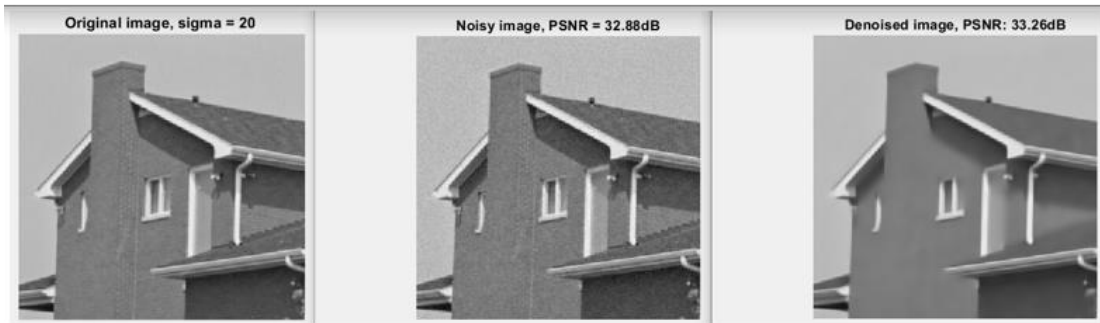
Εικόνα 4.10.α Image\_2 με block size 8, dict size 512 & sigma 10,20,30



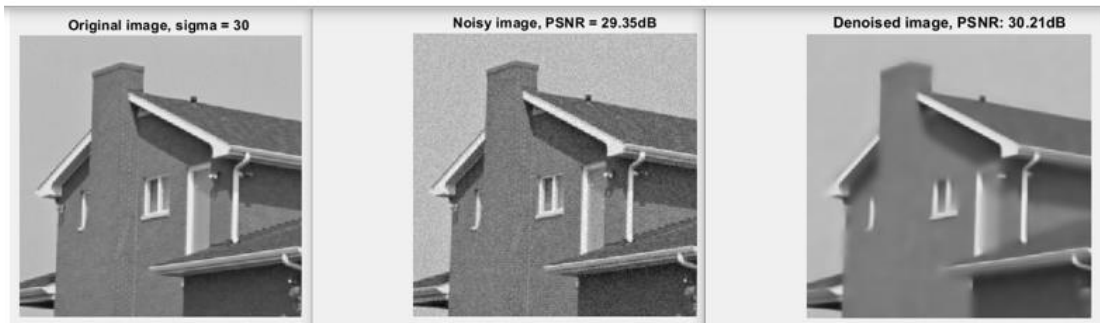
Εικόνα 4.10.b Image\_2 με block size 8, dict size 512 & sigma 10,20,30



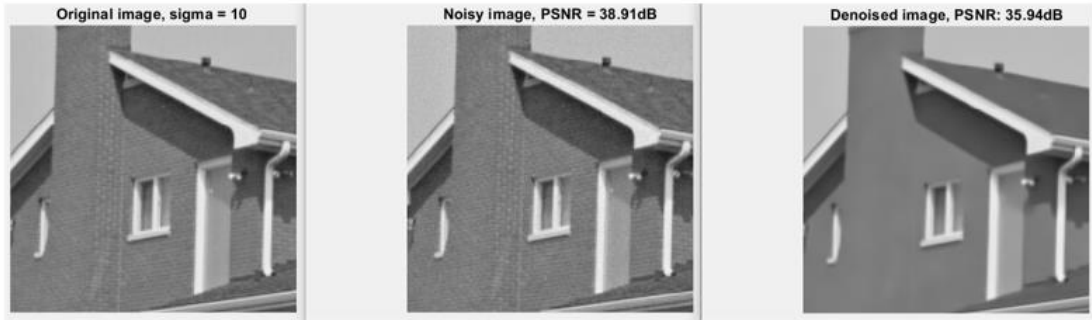
↙



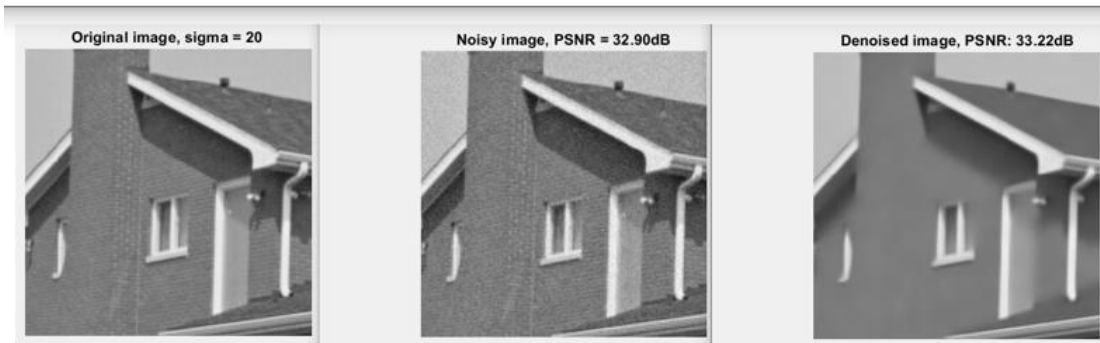
↙



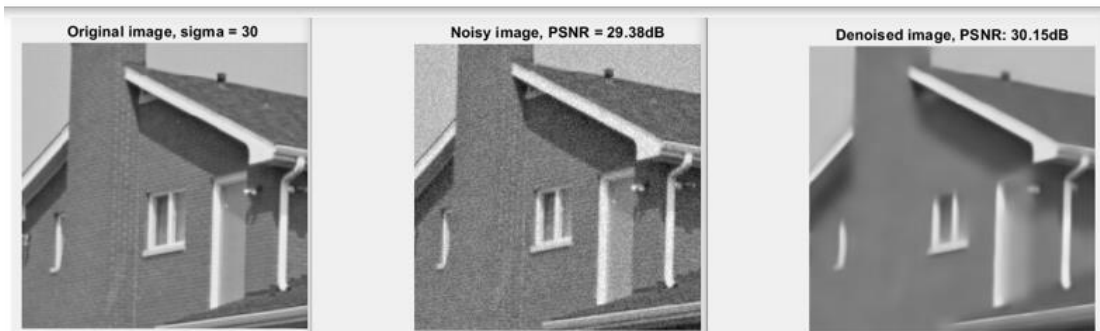
Εικόνα 4.11.α Image\_3 με block size 8, dict size 512 & sigma 10,20,30



κ



κ



Εικόνα 4.11.b Image\_3 με block size 8, dict size 512 & sigma 10,20,30

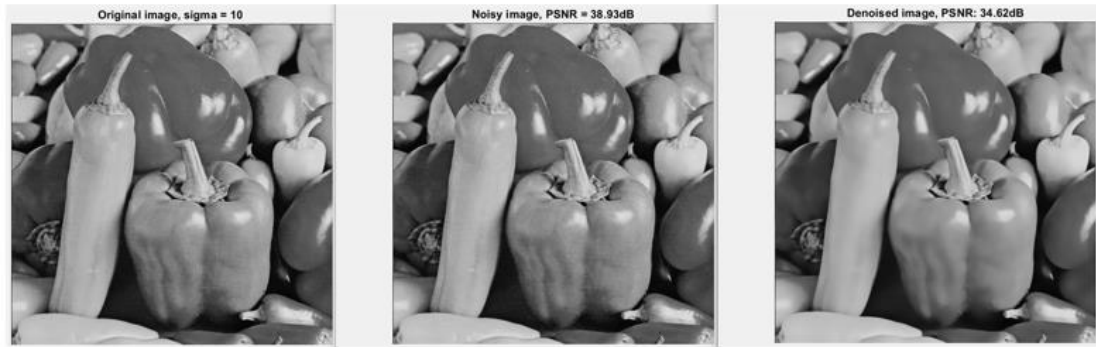




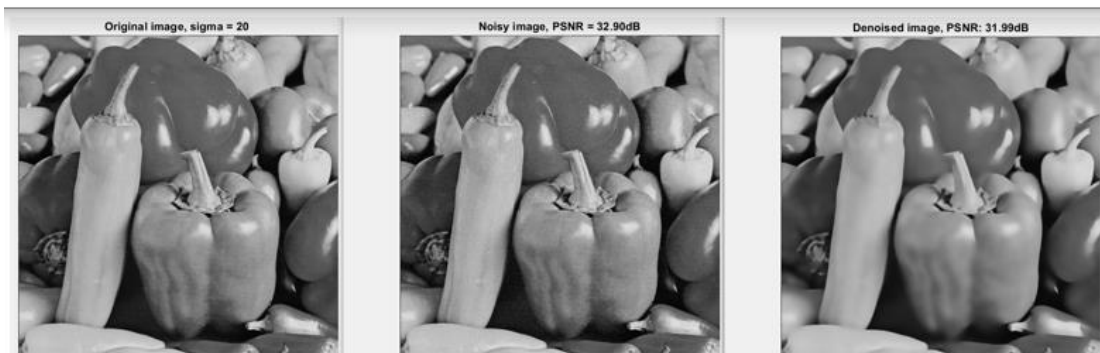
Εικόνα 4.12.α Image\_4 με block size 8, dict size 512 & sigma 10,20,30



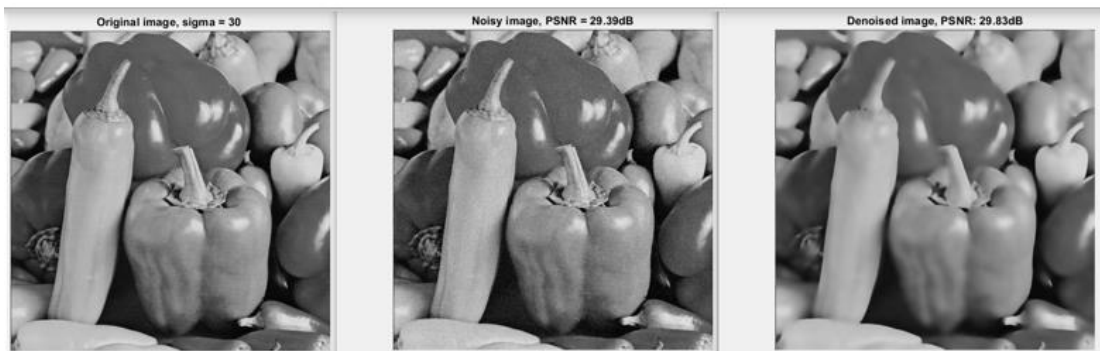
Εικόνα 4.12.b Image\_4 με block size 8, dict size 512 & sigma 10,20,30



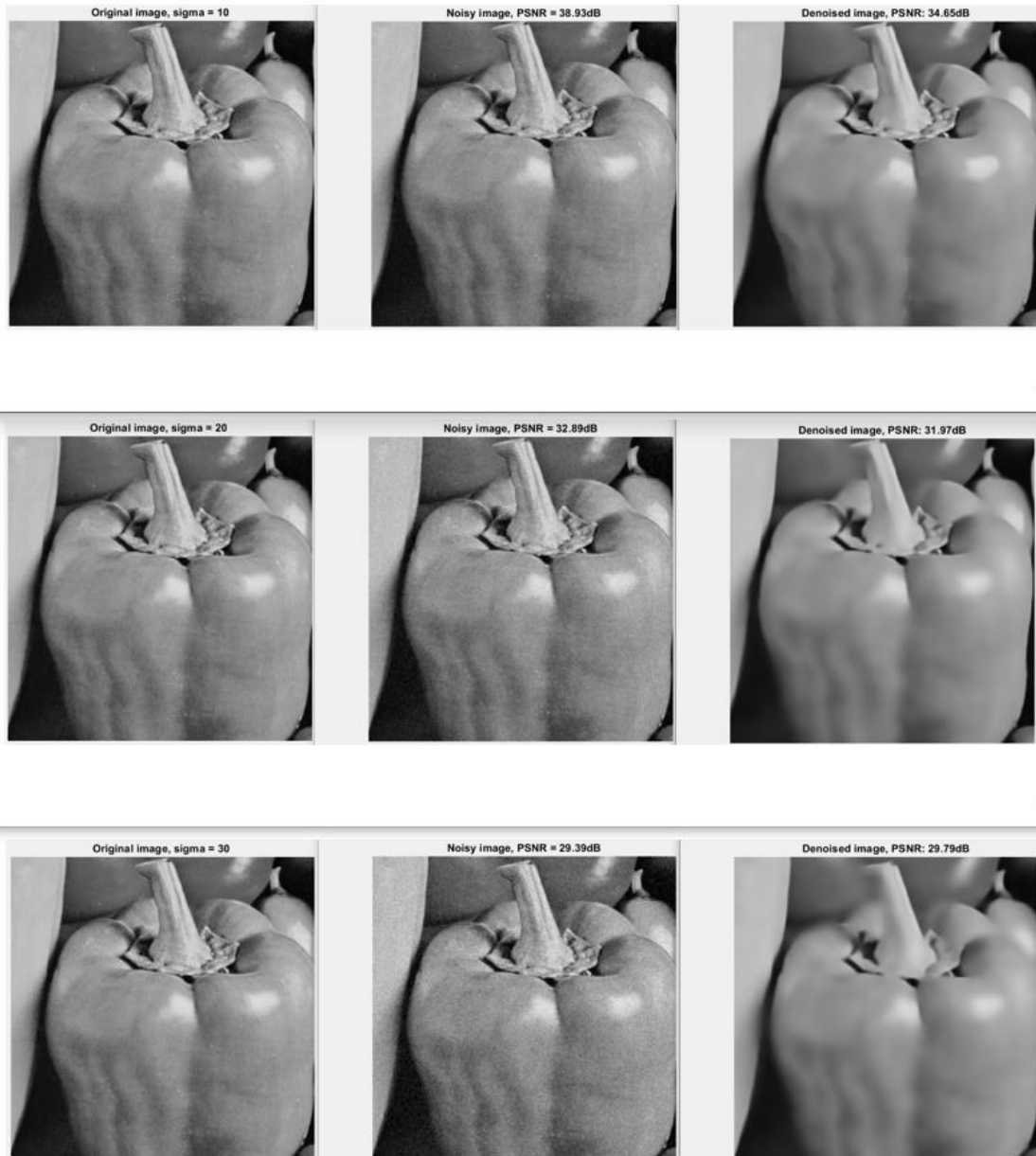
κ



κ



Εικόνα 4.13.α Image\_5 με block size 8, dict size 512 & sigma 10,20,30



Εικόνα 4.13.b Image\_5 με block size 8, dict size 512 & sigma 10,20,30

Στα παραπάνω πειράματα παρατηρούμε ότι η αύξηση της τιμής της τυπικής απόκλισης στον θόρυβο δημιουργεί μεγαλύτερο θάμπωμα (blurred) στην αποθορυβοποιημένη εικόνα. Το φαινόμενο αυτό μπορεί να ερμηνευτεί ως εξής: ο σκοπός της αποθορυβοποίησης είναι να εξαλειφθούν σημεία ή μικρές περιοχές όπου έχει επιδράσει ο θόρυβος αλλάζοντας σε μεγάλο βαθμό την τιμή του εικονοστοιχείου θεωρώντας ότι τα γειτονικά του έχουν πολύ κοντινές τιμές με αυτό. Η επίδραση του θορύβου είναι πιο έντονη στις περιοχές όπου η τιμή του θορυβώδους εικονοστοιχείου αποκλίνει από τα αρχικά γειτονικά του εικονοστοιχεία και με την αποθορυβοποίηση θέλουμε να εξαλειφθούν αυτές οι περιοχές. Έτσι κατά την

αποθροβοποίηση αλλάζουμε τιμές σε γειτονικά εικονοστοιχεία που σημειώνουν μεγάλη απόκλιση ώστε να απέχουν ελάχιστα δημιουργώντας έτσι το φαινόμενο του θαμπώματος της εικόνας.



Εικόνα 4.14 Image\_1 με block size 8, sigma 20 & dict size 64,128,256,512



Εικόνα 4.15 Image\_2 με block size 8, sigma 20 & dict size 64,128,256,512

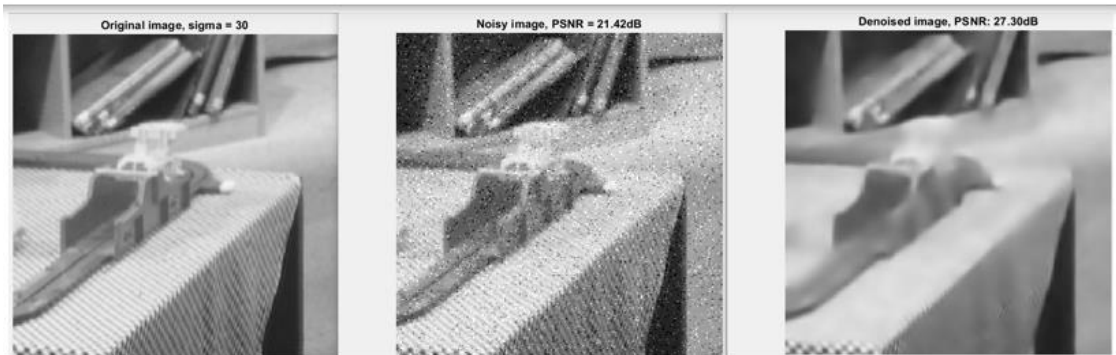
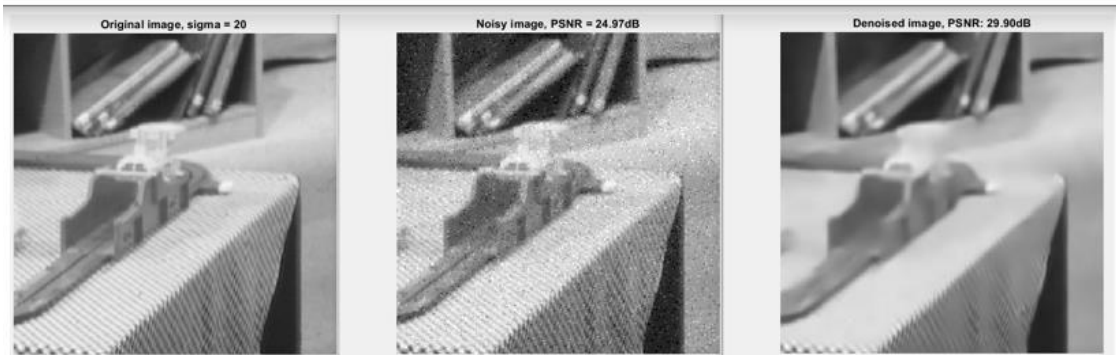
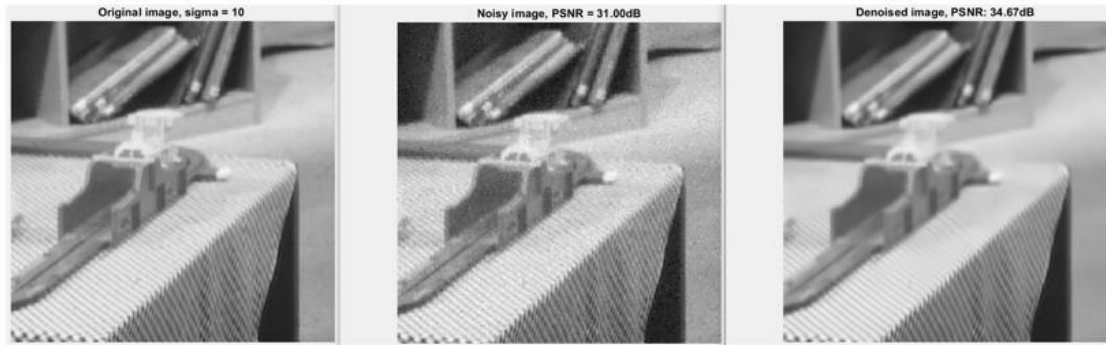
Στην εικόνες 4.14 και 4.15 παρατηρούμε ότι η αλλαγή του dict size δεν έχει επίπτωση στην μετρική PSNR κατά την αποθροβοποίηση μίας εικόνας.

#### 4.5. Αποθρομβοποίηση σήματος με θόρυβο salt n pepper

Ο θόρυβος salt and pepper είναι μία περίπτωση χαρακτηριστική που ανήκει στα προβλήματα του image inpainting αλλά απευθύνεται σε grayscale εικόνες. Στον θόρυβο salt and pepper κάποιο μέρος των pixels της εικόνας αντικαθίστανται με τις ακραίες τιμές της έντασης των pixels. Στις εικόνες που ακολουθούν φαίνεται η αποθρομβοποίηση με salt and pepper.



Εικόνα 4.16.a Image\_1 με block size 8, dict size 128 & sigma 10,20,30

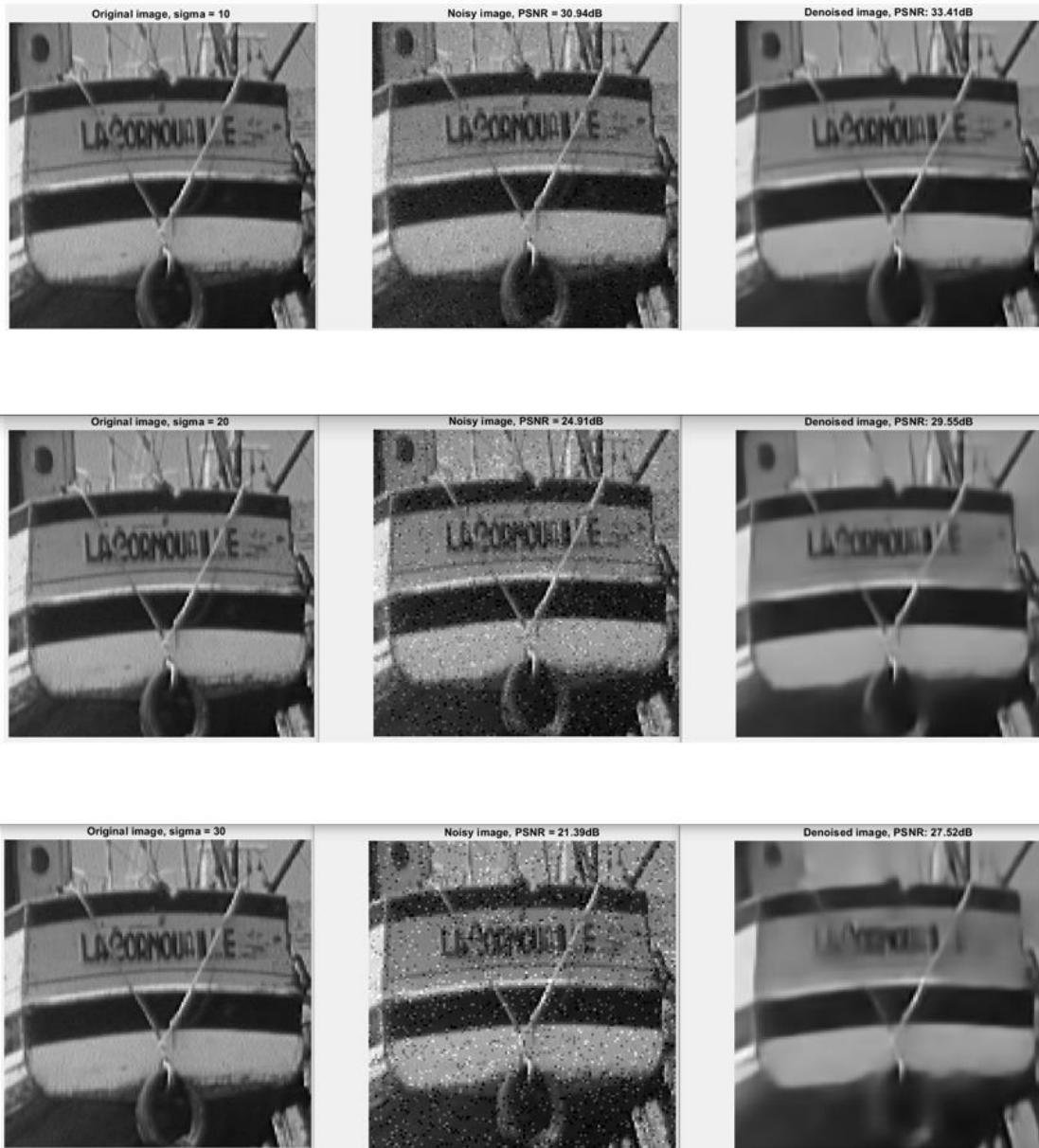


Εικόνα 4.16.b Image\_1 με block size 8, dict size 128 & sigma 10,20,30

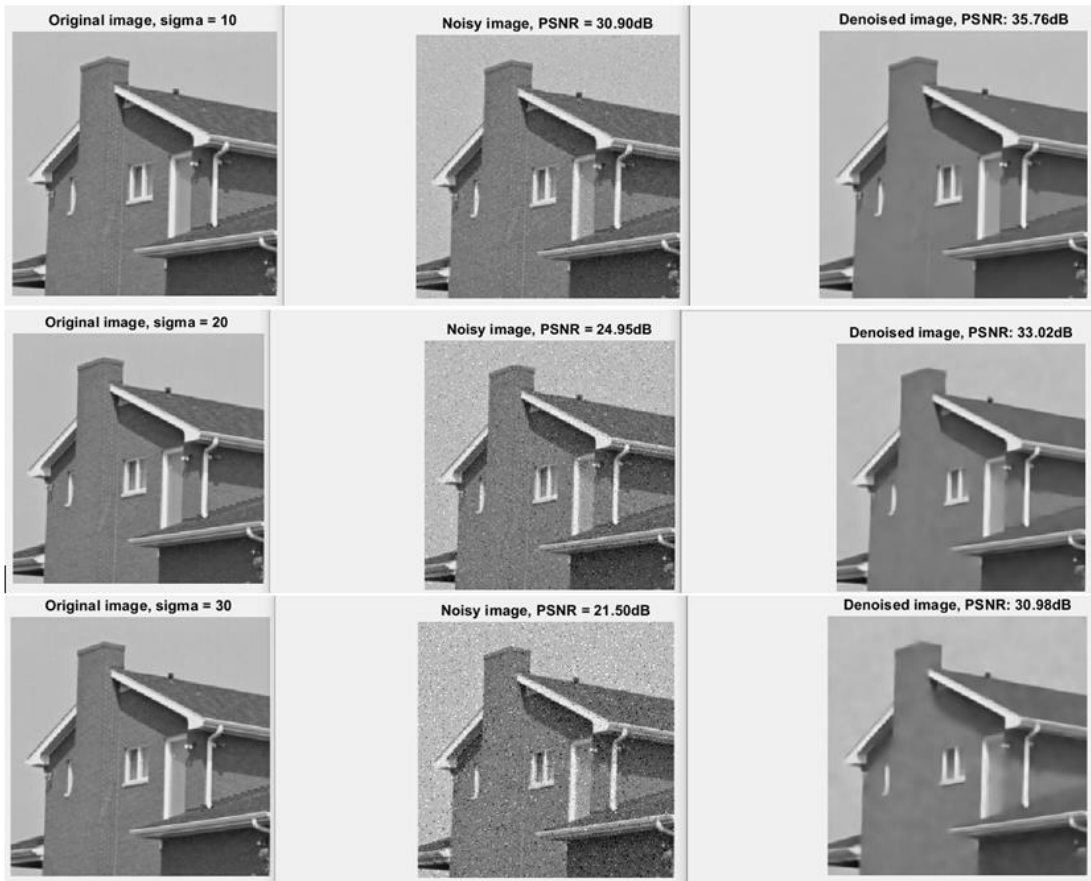




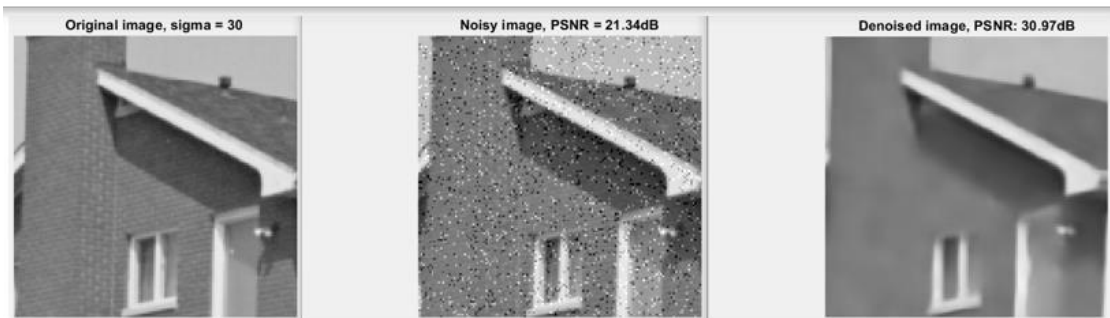
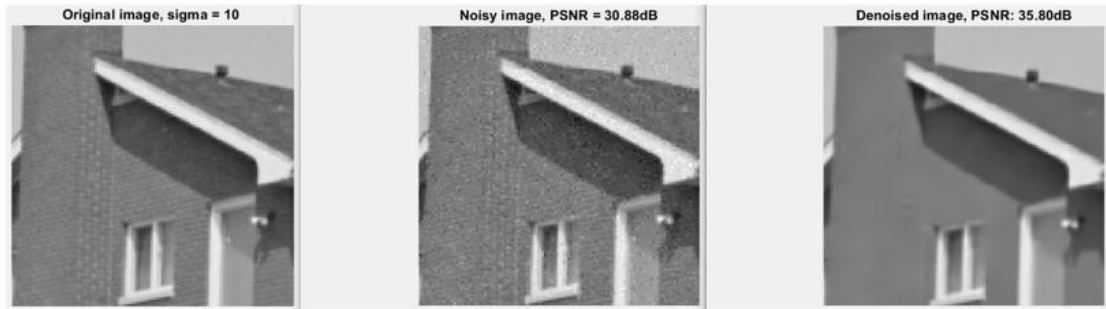
Εικόνα 4.17.α Image\_2 με block size 8 dict size 128 & sigma 10,20,30



Εικόνα 4.17.b Image\_2 με block size 8 dict size 128 & sigma 10,20,30



Εικόνα 4.18.a Image\_3 με block size 8 dict size 128 & sigma 10,20,30



Εικόνα 4.18.b Image\_3 με block size 8 dict size 128 & sigma 10,20,30



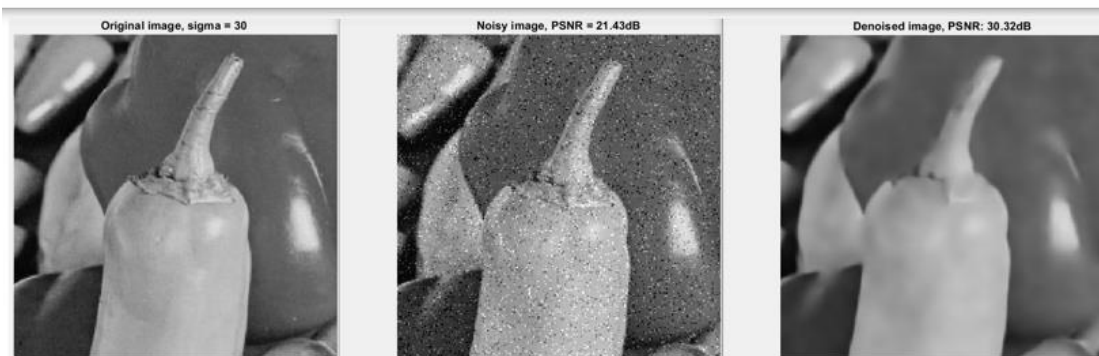
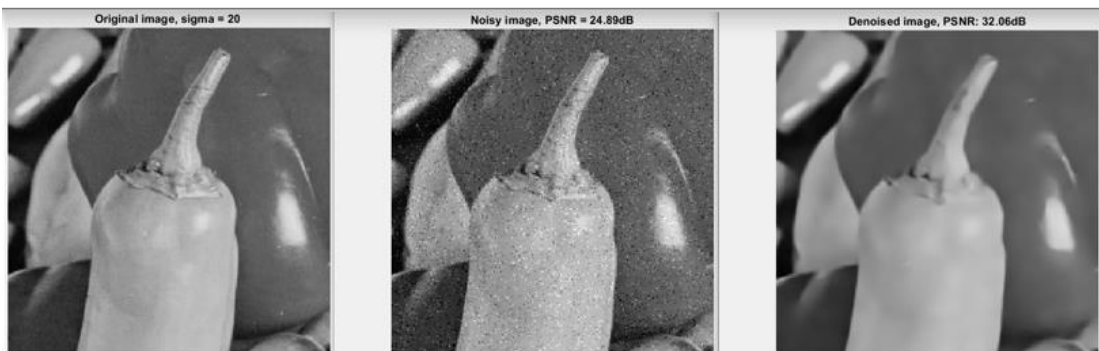
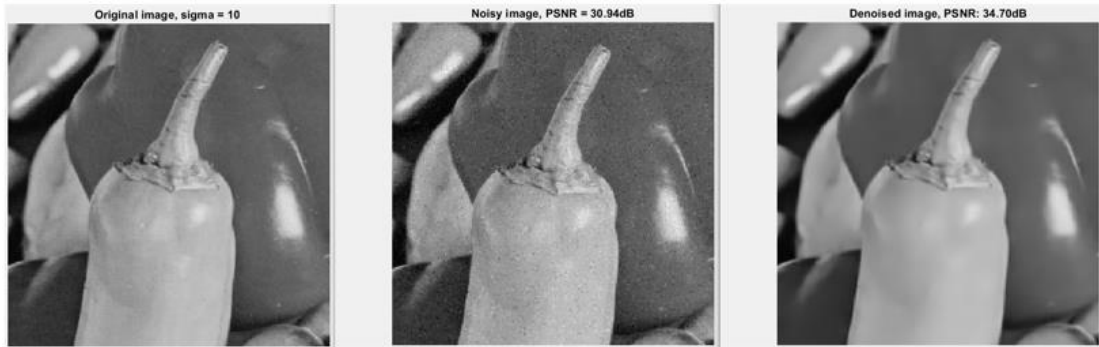
Εικόνα 4.19.α Image\_4 με block size 8 dict size 128 & sigma 10,20,30



Εικόνα 4.19.b Image\_4 με block size 8 dict size 128 & sigma 10,20,30

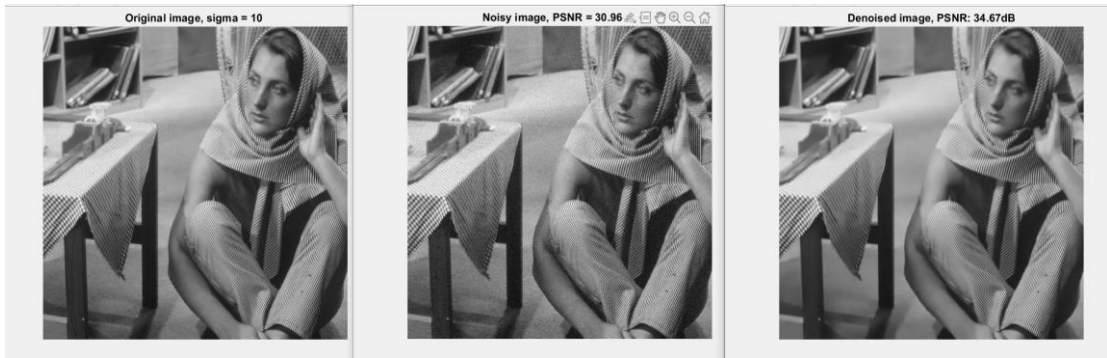


Εικόνα 4.20.α Image\_5 με block size 8 dict size 128 & sigma 10,20,30

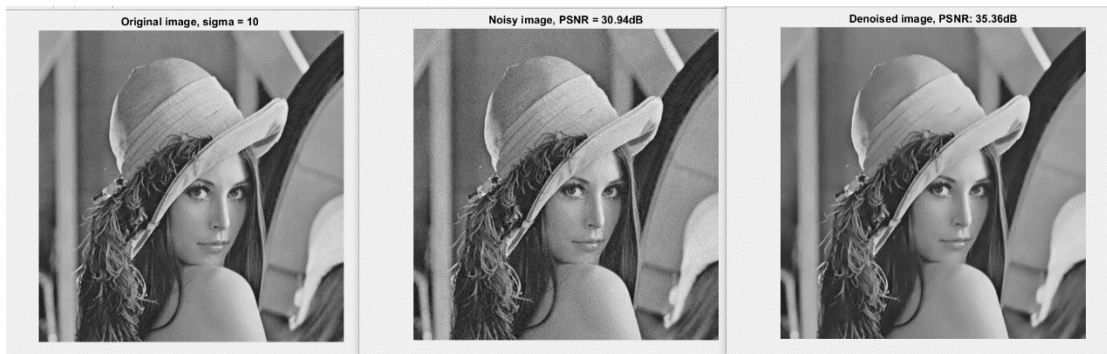


Εικόνα 4.20.b Image\_5 με block size 8 dict size 128 & sigma 10,20,30

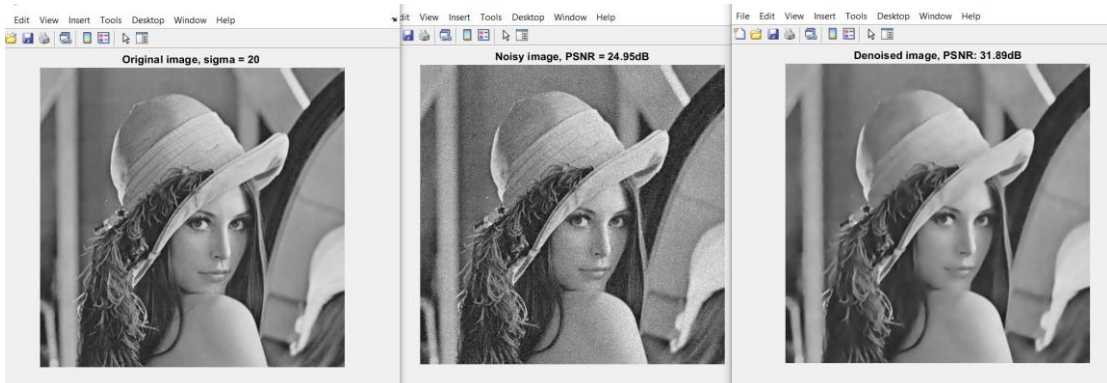




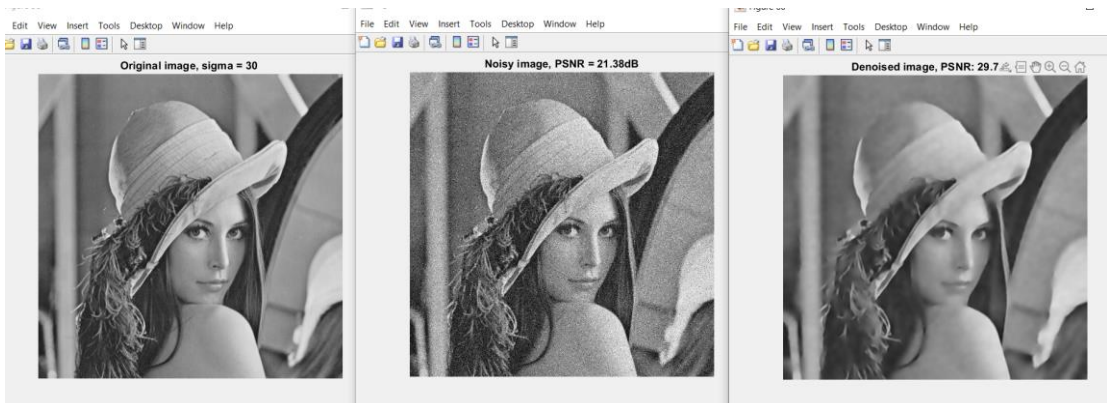
Εικόνα 4.21 Image\_1, με Θόρυβο salt & pepper blocksize 8, dictsize 256 & sigma 10.



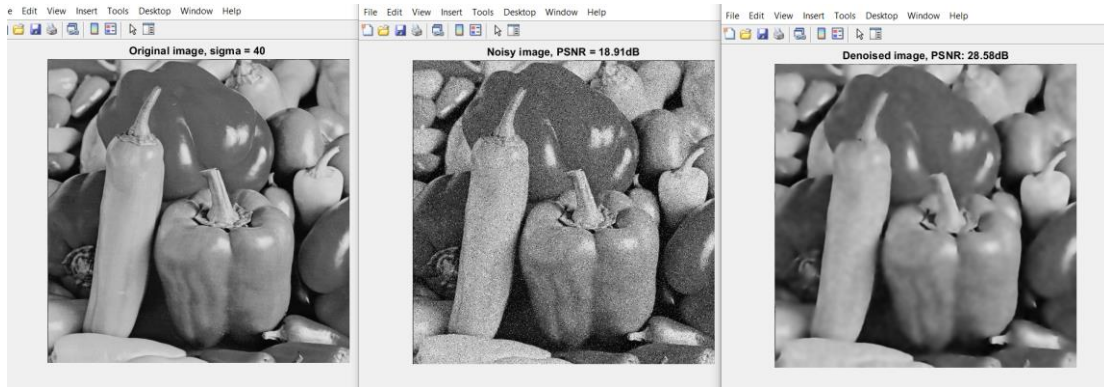
Εικόνα 4.22 Image\_4, με Θόρυβο salt & pepper blocksize 8, dictsize 256 & sigma 10.



Εικόνα 4.23 Image\_4, με Θόρυβο salt & pepper blocksize 8 dictsize 265 & sigma 20.



Εικόνα 4.24 Image\_4, με Θόρυβο salt & pepper, blocksize 8, dictsize 256 & sigma 30.



Εικόνα 4.25 Image\_5, με Θόρυβο salt & pepper, blocksize 8, dictsize 256 & sigma 40.

Στους πίνακες που ακολουθούν φαίνονται τα συγκεντρωτικά αποτελέσματα του προτεινόμενου αλγορίθμου και για τα τρία ήδη θορύβου που μελετήσαμε για όλες τις εικόνες

| test_image | noise_type    | image_size | block_size | dict_size | sigma | PSNR  |
|------------|---------------|------------|------------|-----------|-------|-------|
| image1     | gauss         | 512        | 8          | 512       | 10    | 34,62 |
| image1     | gauss         | 512        | 8          | 512       | 20    | 30,89 |
| image1     | gauss         | 512        | 8          | 512       | 30    | 28,58 |
| image1     | gauss         | 512        | 8          | 512       | 40    | 26,9  |
| image1     | uniform       | 512        | 8          | 512       | 10    | 34,43 |
| image1     | uniform       | 512        | 8          | 512       | 20    | 29,34 |
| image1     | uniform       | 512        | 8          | 512       | 30    | 26,15 |
| image1     | uniform       | 512        | 8          | 512       | 40    | 23,49 |
| image1     | salt_n_pepper | 512        | 8          | 512       | 10    | 35,01 |
| image1     | salt_n_pepper | 512        | 8          | 512       | 20    | 30,25 |
| image1     | salt_n_pepper | 512        | 8          | 512       | 30    | 27,67 |
| image1     | salt_n_pepper | 512        | 8          | 512       | 40    | 25,24 |

Πίνακ 1 PSNR image\_1 για 3 type noise & sigma 10,20,30,40

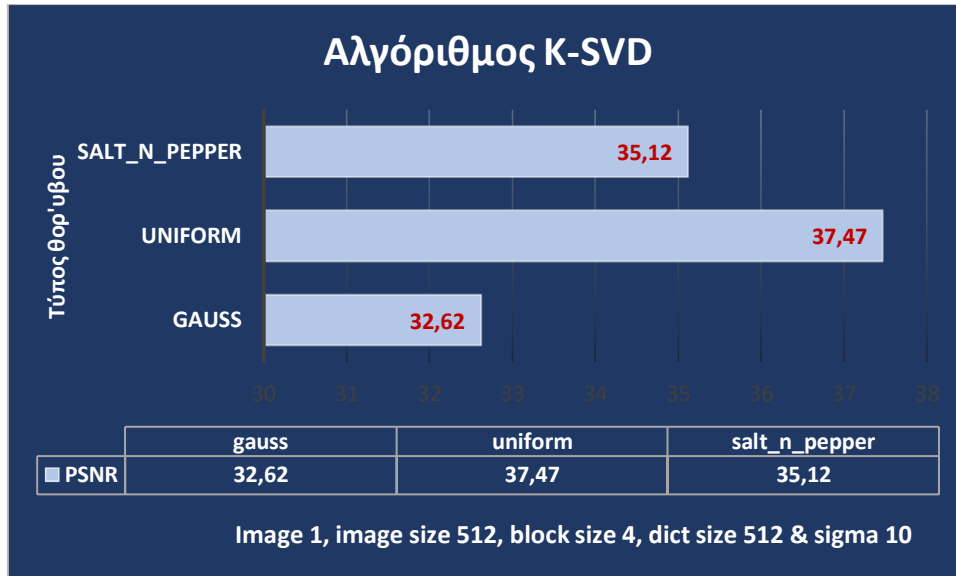
| test_image | noise_type    | image_size | block_size | dict_size | sigma | PSNR  |
|------------|---------------|------------|------------|-----------|-------|-------|
| image1     | gauss         | 512        | 4          | 512       | 20    | 28,26 |
| image1     | gauss         | 512        | 8          | 512       | 20    | 30,89 |
| image1     | uniform       | 512        | 4          | 512       | 20    | 31,86 |
| image1     | uniform       | 512        | 8          | 512       | 20    | 29,35 |
| image1     | salt_n_pepper | 512        | 4          | 512       | 20    | 30,55 |
| image1     | salt_n_pepper | 512        | 8          | 512       | 20    | 30,25 |
| image2     | gauss         | 512        | 4          | 512       | 20    | 28,69 |
| image2     | gauss         | 512        | 8          | 512       | 20    | 30,37 |
| image2     | uniform       | 512        | 4          | 512       | 20    | 32,13 |
| image2     | uniform       | 512        | 8          | 512       | 20    | 29,06 |
| image2     | salt_n_pepper | 512        | 4          | 512       | 20    | 30,71 |
| image2     | salt_n_pepper | 512        | 8          | 512       | 20    | 29,76 |
| image3     | gauss         | 256        | 4          | 512       | 20    | 29,61 |
| image3     | gauss         | 256        | 8          | 512       | 20    | 33,42 |
| image3     | uniform       | 256        | 4          | 512       | 20    | 34,93 |
| image3     | uniform       | 256        | 8          | 512       | 20    | 33,26 |
| image3     | salt_n_pepper | 256        | 4          | 512       | 20    | 32,49 |
| image3     | salt_n_pepper | 256        | 8          | 512       | 20    | 33,69 |
| image4     | gauss         | 512        | 4          | 512       | 20    | 29,66 |
| image4     | gauss         | 512        | 8          | 512       | 20    | 32,39 |
| image4     | uniform       | 512        | 4          | 512       | 20    | 34,25 |
| image4     | uniform       | 512        | 8          | 512       | 20    | 31,5  |
| image4     | salt_n_pepper | 512        | 4          | 512       | 20    | 32,22 |
| image4     | salt_n_pepper | 512        | 8          | 512       | 20    | 32,09 |
| image5     | gauss         | 512        | 4          | 512       | 20    | 29,72 |
| image5     | gauss         | 512        | 8          | 512       | 20    | 32,32 |
| image5     | uniform       | 512        | 4          | 512       | 20    | 34,52 |
| image5     | uniform       | 512        | 8          | 512       | 20    | 31,96 |
| image5     | salt_n_pepper | 512        | 4          | 512       | 20    | 32,35 |
| image5     | salt_n_pepper | 512        | 8          | 512       | 20    | 32,3  |

Πίνακας 2 PSNR για όλα τα Image με block size 4,8

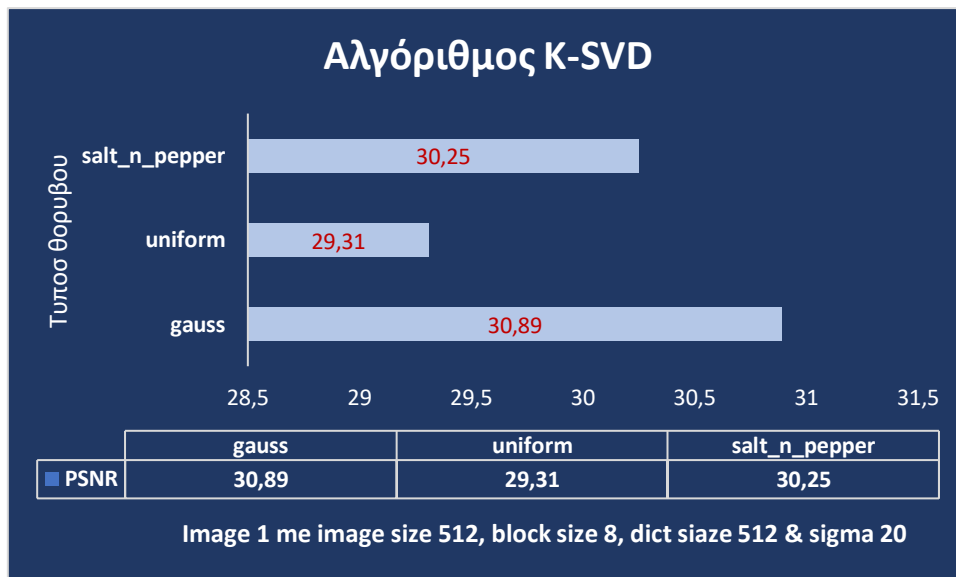
| test_image | noise_type    | image_size | block_size | dict_size | sigma | PSNR  |
|------------|---------------|------------|------------|-----------|-------|-------|
| image1     | gauss         | 512        | 4          | 512       | 20    | 28,26 |
| image1     | gauss         | 512        | 4          | 256       | 20    | 28,5  |
| image1     | uniform       | 512        | 4          | 512       | 20    | 31,88 |
| image1     | uniform       | 512        | 4          | 256       | 20    | 31,75 |
| image1     | salt_n_pepper | 512        | 4          | 512       | 20    | 30,55 |
| image1     | salt_n_pepper | 512        | 4          | 256       | 20    | 30,55 |
| image2     | gauss         | 512        | 4          | 512       | 20    | 28,69 |
| image2     | gauss         | 512        | 4          | 256       | 20    | 28,87 |
| image2     | uniform       | 512        | 4          | 512       | 20    | 32,13 |
| image2     | uniform       | 512        | 4          | 256       | 20    | 32,06 |
| image2     | salt_n_pepper | 512        | 4          | 512       | 20    | 30,71 |
| image2     | salt_n_pepper | 512        | 4          | 256       | 20    | 30,82 |
| image3     | gauss         | 256        | 4          | 512       | 20    | 29,61 |
| image3     | gauss         | 256        | 4          | 256       | 20    | 29,99 |
| image3     | uniform       | 256        | 4          | 512       | 20    | 35    |
| image3     | uniform       | 256        | 4          | 256       | 20    | 34,99 |
| image3     | salt_n_pepper | 256        | 4          | 512       | 20    | 32,49 |
| image3     | salt_n_pepper | 256        | 4          | 256       | 20    | 32,58 |
| image4     | gauss         | 512        | 4          | 512       | 20    | 29,66 |
| image4     | gauss         | 512        | 4          | 256       | 20    | 29,92 |
| image4     | uniform       | 512        | 4          | 512       | 20    | 34,25 |
| image4     | uniform       | 512        | 4          | 256       | 20    | 34,24 |
| image4     | salt_n_pepper | 512        | 4          | 512       | 20    | 32,22 |
| image4     | salt_n_pepper | 512        | 4          | 256       | 20    | 32,41 |
| image5     | gauss         | 512        | 4          | 512       | 20    | 29,72 |
| image5     | gauss         | 512        | 4          | 256       | 20    | 29,89 |
| image5     | uniform       | 512        | 4          | 512       | 20    | 34,52 |
| image5     | uniform       | 512        | 4          | 256       | 20    | 34,51 |
| image5     | salt_n_pepper | 512        | 4          | 512       | 20    | 32,35 |
| image5     | salt_n_pepper | 512        | 4          | 256       | 20    | 32,48 |

Πίνακας 3 PSNR για όλα τα image με dict size 256, 512

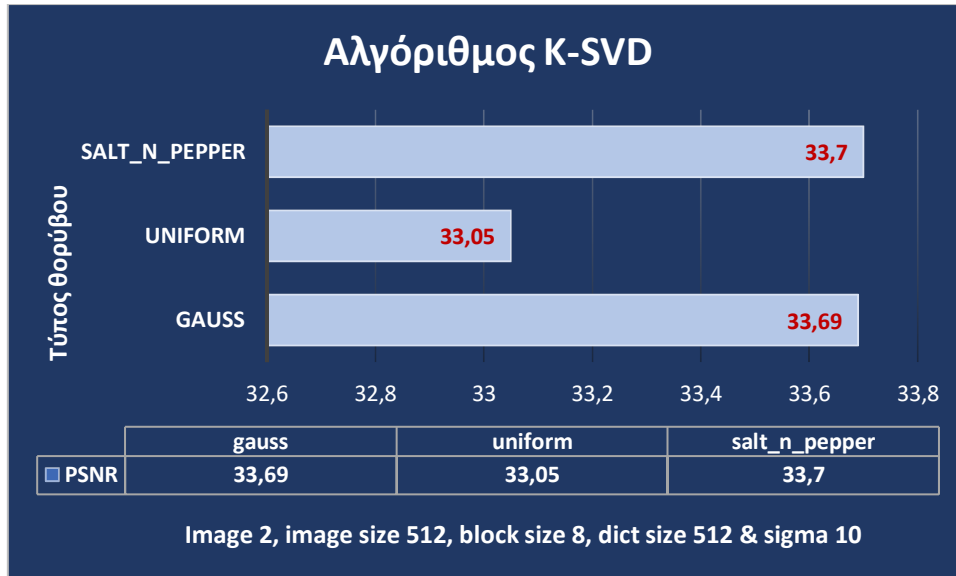
Επίσης, παρουσιάζονται και αντίστοιχα διαγράμματα που αποτυπώνουν για κάθε εικόνα και για κάθε τύπο θορύβου που μελετάμε την απόδοση της μετρικής PSNR κατά την εφαρμογή του αλγορίθμου k-SVD.



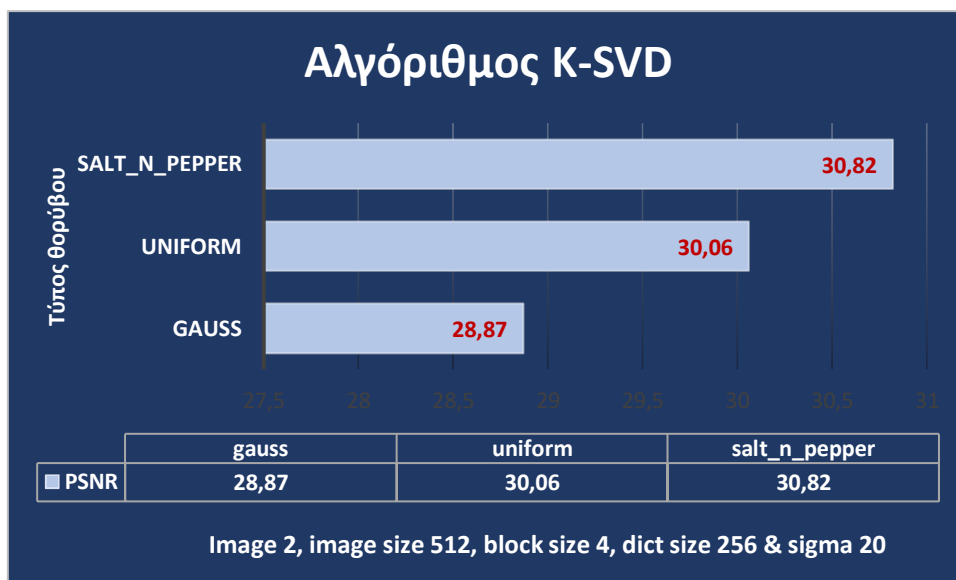
Σχήμα 8: Image\_1 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 10



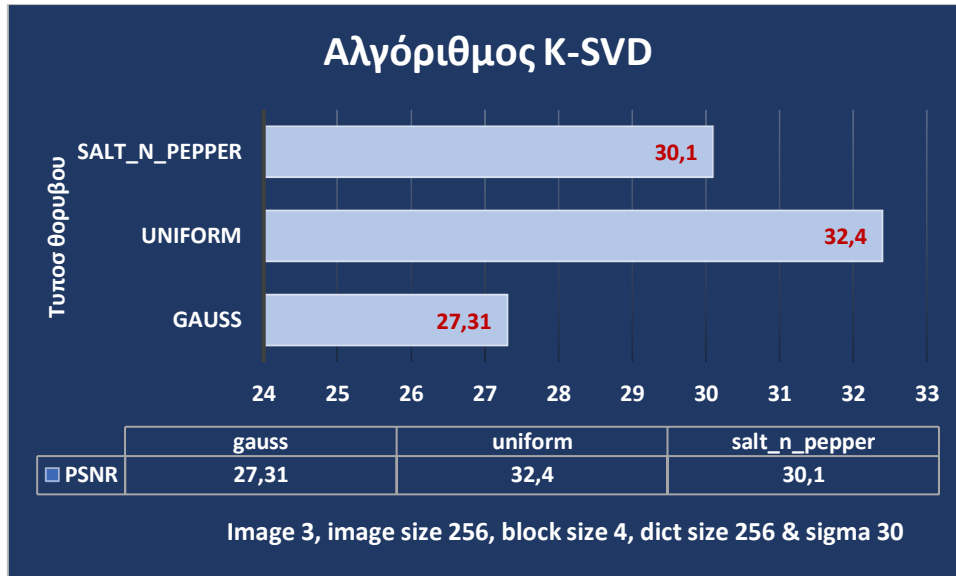
Σχήμα 9: Image\_1 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 20



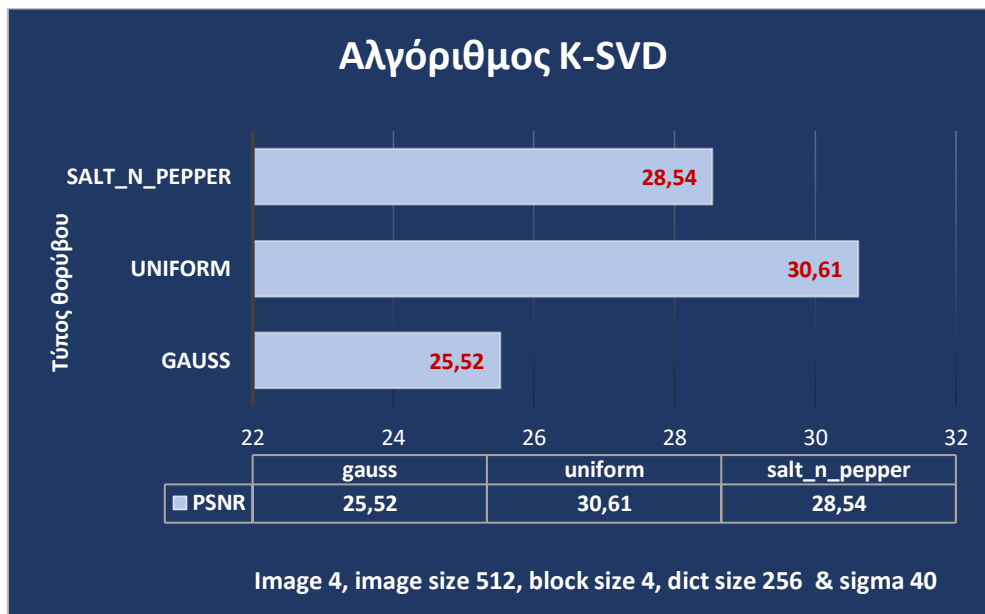
Σχήμα 10: Image\_2 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 10



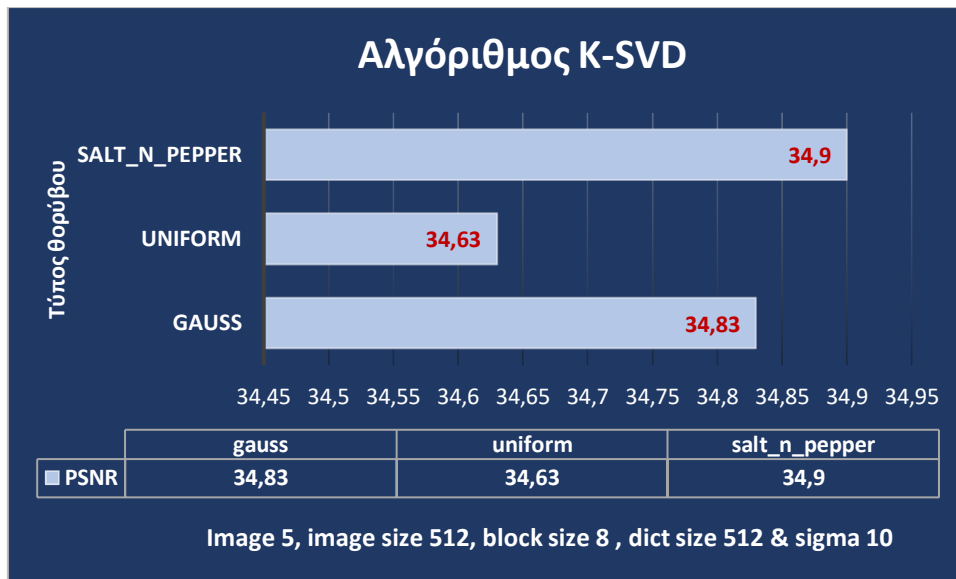
Σχήμα 11: Image\_2 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 20



Σχήμα 12: Image\_3 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 30

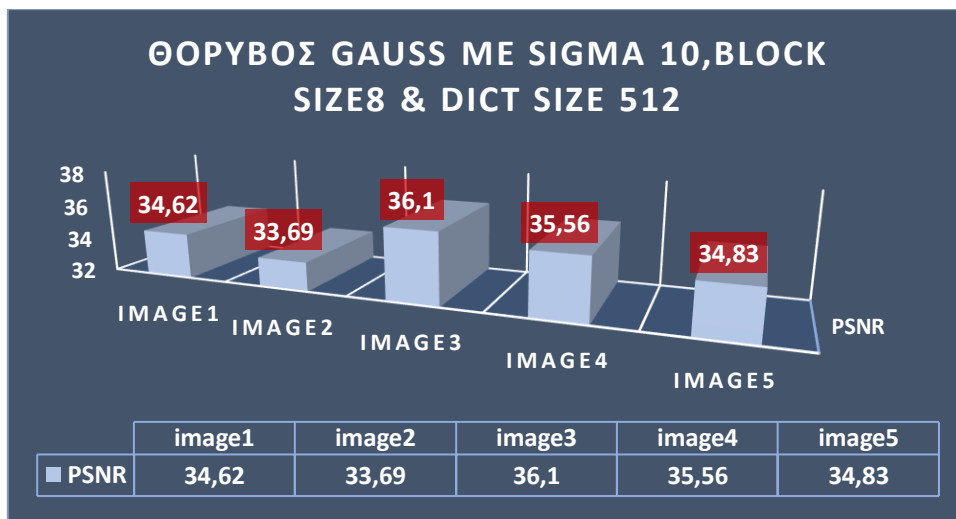


Σχήμα 13: Image\_4 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 40



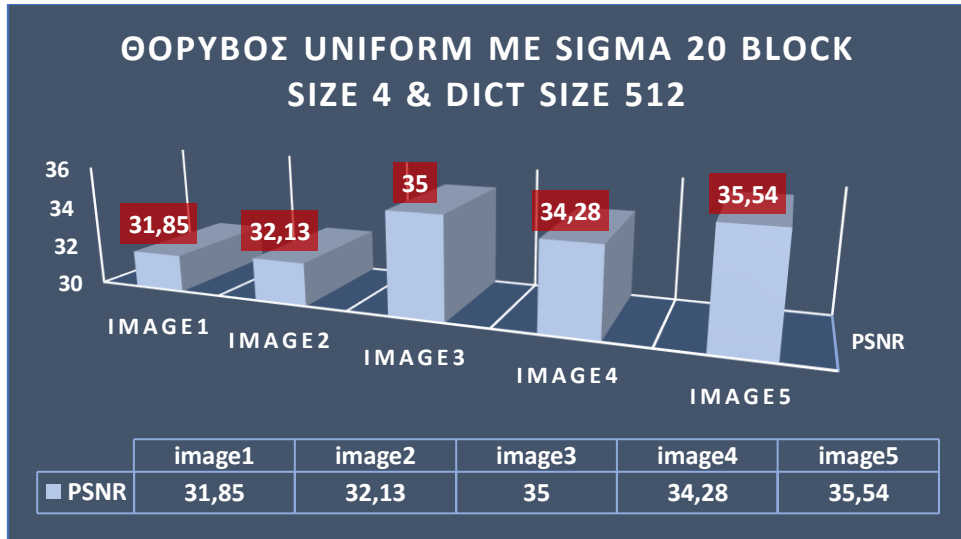
Σχήμα 14: Image\_5 Ο αλγόριθμος k-SVD με 3 διαφορετικούς τύπους θορύβου & sigma 10

Εκτελώντας αρκετές φορές τον αλγόριθμο και για ποικίλα παραμέτρων παρατηρήσαμε ότι όσο μεγαλύτερο PSNR έχουμε σαν μετρική στην αποθρομβοποίηση της εικόνας μας κάθε φορά τόσο καλύτερο το αποτέλεσμα αυτής. Αυτό ισχύει για όλους τους τύπους θορύβου που εφαρμόζουμε και για όλες τις εικόνες.

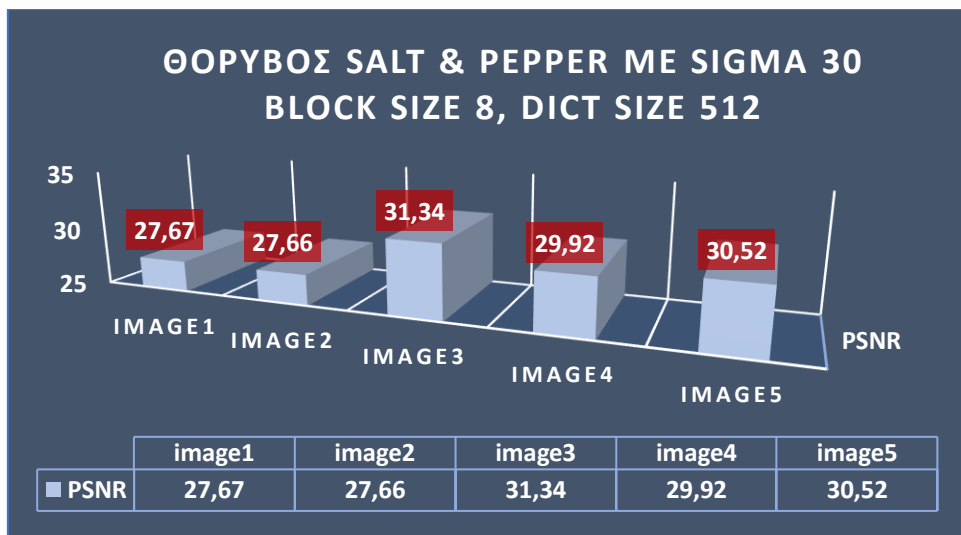


Σχήμα 15: Σύγκριση 5 εικόνων με θόρυβο Gauss & sigma 10





Σχήμα 16: Σύγκριση 5 εικόνων με θόρυβο Ομοιόμορφης Κατανομής & sigma 20



Σχήμα 17: Σύγκριση 5 εικόνων με θόρυβο salt and pepper & sigma 30

## Κεφάλαιο 5<sup>ο</sup> – ΑΝΑΚΑΤΑΣΚΕΥΗ ΤΜΗΜΑΤΩΝ ΕΙΚΟΝΑΣ

### 5.1 Γενικά

Η τεχνική που προσπαθεί να καλύψει τα σημεία που λείπουν από μια εικόνα ή που σε ένα βαθμό είναι κατεστραμμένα ονομάζεται *retouching* or *inpainting*. [23]

Αναλυτικότερα η τεχνική του *image inpainting* αφορά τη διαδικασία ανακατασκευής τμημάτων ή αφαίρεσης τμημάτων μίας εικόνας τα οποία έχουν τροποποιηθεί ή αλλοιωθεί. Στόχος είναι το τελικό αποτέλεσμα να είναι οπτικά ρεαλιστικό ώστε να μην φαίνεται όσο το δυνατό η παρέμβαση που έχει γίνει. Στον τομέα της μηχανικής όρασης των υπολογιστών για την τεχνική του *image inpainting* υπάρχουν δύο προσεγγίσεις: κατά την πρώτη προσέγγιση στις εν λόγω περιοχές τοποθετούμε μικρά αντίγραφα γειτονικών περιοχών, ενώ κατά την δεύτερη χρησιμοποιούμε προστροφοδοτικά μοντέλα τα οποία στηρίζουν τη λειτουργία τους σε νευρωνικά δίκτυα (συνελικτικά δίκτυα μηχανικής μάθησης). [26]

Στις μεθόδους που έχουμε ασχοληθεί μέχρι τώρα ο αλγόριθμος *k-SVD* έχει στόχο να διορθώσει τις τιμές των εικονοστοιχείων μιας εικόνας στα οποία έχει επιδράσει ο θόρυβος. Υπάρχει όμως περίπτωση η τιμή του σήματος μας να μην είναι γνωστή σε όλο το εύρος της. Για να εντοπίσουμε τιμές που δεν είναι γνωστές εφαρμόζουμε αλγόριθμους που ο τρόπος που δουλεύουν προσεγγίζει αυτόν της αριθμητικής παρεμβολής. Αναλυτικότερα με βάση τις δεδομένες τιμές προσεγγίζουν τις άγνωστες τιμές του σήματος. Σε περίπτωση που έχουμε ψηφιακή εικόνα η προσέγγιση της τιμής της έντασης των *pixel* που δεν γνωρίζουμε προσδιορίζεται μέσω τεχνικών *image inpainting* [8].

Το χρώμα είναι ένας από τους πιο σημαντικούς παράγοντες περιγραφής που συνήθως απλουστεύει σε μεγάλο βαθμό την διαδικασία αναγνώρισης αντικειμένων και την εξαγωγή περιοχών. Τον όρο *inpainting* τον πήραμε από τον καλλιτεχνικό κόσμο αρχικά στη επεξεργασία εικόνας οι ερευνητές τον αποκαλούσαν παρεμβολή (*interpolation*). Πρώτοι οι Bertalmi, Sapiro, Caselles και Ballester ανέφεραν εκτενέστερα τα κίνητρα για να ασχοληθούμε με το πρόβλημα του *image inpainting* καθώς αυτοί ήταν που καθιέρωσαν και τον όρο αυτό στην επεξεργασία εικόνας [25].

Υπάρχουν πολλές μέθοδοι που σχεδιάστηκαν για την επεξεργασία μονόχρωμης εικόνας οι οποίες μπορούν να επεκταθούν και αν μελετάμε διανυσματική περίπτωση δηλαδή έγχρωμη εικόνα. Συχνά όμως δεν έχουμε το ίδιο καλό αποτέλεσμα στην διανυσματική περίπτωση. Αυτό οφείλεται κυρίως στην εξάρτηση που έχουν οι συνιστώσες καθώς και στη μη γραμμική αντίληψη των χρωμάτων.

## 5.2 Ελλιπή δεδομένα εικόνων – αφαίρεση γραμμάτων

Ένα θέμα που προκύπτει στην ανάλυση ψηφιακών είναι αυτό των ελλιπών ή φθαρμένων τμημάτων των εικόνων όπως για παράδειγμα μιας φωτογραφίας με γρατζουνιές. Υπάρχει μία προσέγγιση για την επίλυση αυτού του προβλήματος και αυτού του διαχωρισμού δεδομένων. Δηλαδή ελαχιστοποίηση του κανόνα των συντελεστών ανάλυσης σε σχέση με τα πλαίσια (frames). Το image inpainting είναι μία τεχνική για να καλύψουμε χαμένες ή κατεστραμμένες περιοχές από μία εικόνα χρησιμοποιώντας τις διαθέσιμες πληροφορίες από τη γύρω περιοχή. Αυτή η τεχνική μπορεί να έχει ευρύτερη εφαρμογή όπως η αποκατάσταση και η συμπίεση εικόνας. Η παραδοσιακή προσέγγιση βασίζεται στη μέθοδο Partial Differential Equation (PDE). Η μέθοδος αυτή επικεντρώνεται κυρίως στο μέγεθος της περιοχής στόχου που πρέπει να συμπληρώσουμε. Η μέθοδος PDE είναι προσανατολισμένη στα pixel της εικόνας και λειτουργεί καλά εάν η περιοχή που θα συμπληρωθεί είναι μικρή. [24] Οι Qingrei Wang, Donghai Bao, Huang Bai, Liping Chang, Sheng Li and Hong Xu [27] πρότειναν έναν βελτιστοποιημένο αλγόριθμο k-SVD για την εφαρμογή image inpainting σε εικόνες με σκοπό να επιδιορθώσουμε άγνωστα δεδομένα από κατεστραμμένες περιοχές με στόχο να κάνει ολόκληρη την εικόνα οπτικά ομοιόμορφη. Η αραιή αναπαράσταση εικόνων είναι μία από τις βασικές προσεγγίσεις για την επεξεργασία εικόνων .

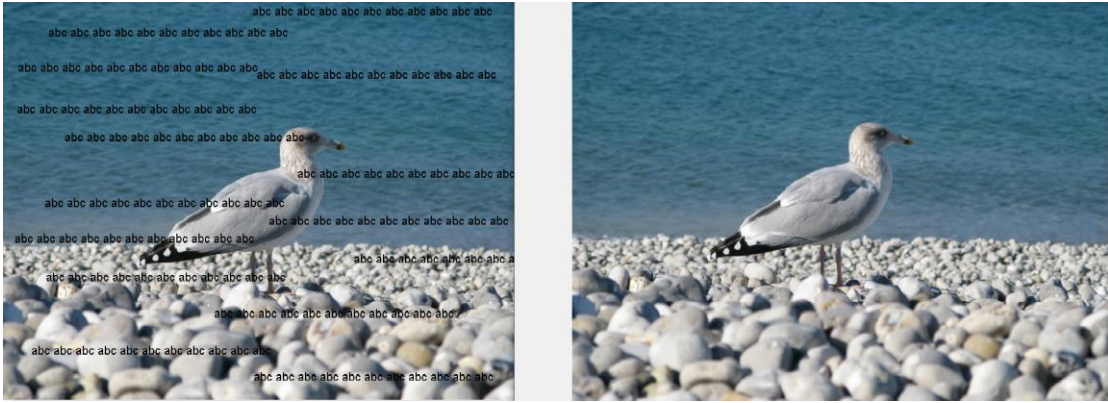
Στην συνέχεια παρουσιάζουμε κάποια από τα αποτελέσματα του αλγορίθμου image inpainting που εφαρμόσαμε σε διαφορετικές εικόνες έγχρωμες και μη, με σκοπό την αποκατάσταση τους.



Εικόνα 5.1 *Image\_6* Inpainting με  $k$ -SVD σε έγχρωμη εικόνα (αριστερά αρχική εικόνα, δεξιά αποτέλεσμα αλγορίθμου)



Εικόνα 5.2 *Image\_1* Inpainting με  $k$ -SVD σε grayscale εικόνα (αριστερά αρχική εικόνα, δεξιά αποτέλεσμα αλγορίθμου)



Εικόνα 5.3 Image\_7 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα (αριστερά αρχική εικόνα με γράμματα, δεξιά αποτέλεσμα αλγορίθμου)  $bb=8,overlap=3,J=3$



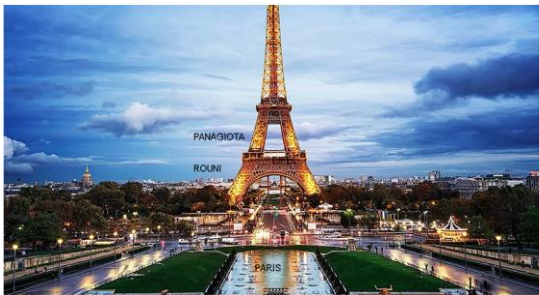
Εικόνα 5.4 Image\_7 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα(αριστερά αρχική εικόνα με γράμματα, δεξιά αποτέλεσμα αλγορίθμου)  $bb=8,overlap=3,J=3$



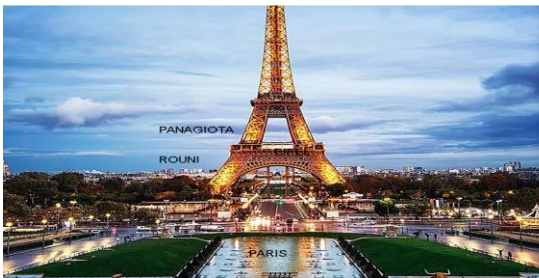
Εικόνα 5.5 Image\_7 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα(αριστερά αρχική εικόνα με γράμματα, δεξιά αποτέλεσμα αλγορίθμου)  $bb=8,overlap=3,J=3$



Εικόνα 5.6 Image\_6 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=3, J=3$



Εικόνα 5.7 Image\_6 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=3, J=3$



Εικόνα 5.8 Image\_6 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=3, J=3$



Εικόνα 5.9 Image\_6 Inpainting με  $k$ -SVD σε έγχρωμη εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=2, J=2$



Εικόνα 5.10 Image\_1 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=3, J=3$



Εικόνα 5.11 Image\_2 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=3, J=3$



Εικόνα 5.12 Image\_3 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, \text{overlap}=2, J=2$

Παρατηρούμε ότι όσο μειώνουμε την τιμή του  $overlap$  και του  $j$ , έστω και κατά μία μονάδα τα αποτελέσματά μας αρχίζουν να μην είναι τόσο ικανοποιητικά. Επίσης όσο αυξάνουμε την τιμή του  $overlap$  και του  $j$  αυξάνεται αντίστοιχα και η υπολογιστική ισχύς που απαιτείται για την εκτέλεση του αλγορίθμου μας.



Εικόνα 5.13 Image\_4 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, overlap=3, J=3$



Εικόνα 5.14 Image\_4 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, overlap=2, J=2$



Εικόνα 5.15 Image\_5 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, overlap=3, J=3$





Εικόνα 5.16 Image\_5 Inpainting με  $k$ -SVD σε grayscale εικόνα αφαίρεση τμημάτων,  $bb=8, overlap=3, J=3$

Ύστερα από πληθώρα δοκιμών καταλήγουμε πως ο αλγόριθμός  $k$ -SVD έχει πολύ καλά αποτελέσματα στην ανακατασκευή και ανάκτηση χαμένης πληροφορίας μιας εικόνας. Επίσης ο αλγόριθμος μας δίνει ικανοποιητικά αποτελέσματα όταν προσπαθεί να αφαιρέσει τα έντονα γράμματα από μια εικόνα, ενώ όσο πιο μικρό είναι τμήμα ή τα γράμματα που θέλουμε να αφαιρέσουμε τόσο καλύτερα είναι τα αποτελέσματα του αλγορίθμου όπως φαίνεται και στις εικόνες που προηγήθηκαν. Επιπλέον καταλήγουμε μέσω των δοκιμών μας πως αλγόριθμος δίνει τα ίδια καλά αποτελέσματα και στην αφαίρεση γραμμάτων ή γραμμών καθώς και στην ανάκτηση πληροφορίας από πολύ «χαλασμένη» εικόνα που είδαμε στις εικόνες 5.1 και 5.2 και αυτό συμβαίνει σε όποιο είδος και μέγεθος εικόνας και αν έχουμε, δηλαδή είτε σε grayscale είτε σε έγχρωμη εικόνα η απόδοση του αλγορίθμου μας είναι εξίσου ικανοποιητική.

## Κεφάλαιο 6<sup>ο</sup> - ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με τεχνικές αραιής κωδικοποίησης και επεξεργασίας εικόνας με τη χρήση του αλγόριθμου k-SVD. Ειδικότερα ύστερα από βιβλιογραφική μελέτη επικεντρωθήκαμε στη αποθρομβοποίηση εικόνων στις οποίες γνωρίζαμε εκ των προτέρων το είδος του θορύβου που είχε υπεισέλθει και εστιάσαμε στην ικανότητα του αλγορίθμου k-SVD για την αποθρομβοποίησή τους. Μελετήσαμε τον αλγόριθμο εκμάθησης λεξικών με την χρήση του k-SVD ο οποίος αποτελεί μία ευέλικτη διαδικασία που σε συνδυασμό με την εκπαίδευση ενός υπερπλήρους λεξικού ανταποκρίνεται καλύτερα σε ένα σύνολο δεδομένων σημάτων.

Έπειτα από σημαντικό πλήθος δοκιμών σε εικόνες συμπεραίνουμε ότι ο αλγόριθμος λειτουργεί ικανοποιητικά όταν εφαρμόζεται τόσο σε grayscale εικόνες όσο και σε έγχρωμες. Στη συνέχεια συγκεντρώνουμε για κάθε εικόνα τις τιμές του PSNR όπως προέκυψαν για κάθε είδος θορύβου που μελετήσαμε και διαπιστώσαμε ότι ο αλγόριθμος δίνει καλά αποτελέσματα και για τους τρεις τύπους θορύβου που μελετήσαμε. Επίσης εκτελώντας τον αλγόριθμο παρατηρήσαμε πως για blocksize 32 ο χρόνος εκτέλεσης ήταν πολύ μεγαλύτερος σε σχέση με τον αντίστοιχη εκτέλεση του αλγορίθμου για blocksize 8 ή 16 χωρίς να δίνει σημαντικά καλύτερα αποτελέσματα στην αποθρομβοποίηση της εικόνας. Ύστερα από αρκετές επαναληπτικές εκτελέσεις του αλγορίθμου και για πληθώρα παραμέτρων παρατηρήσαμε ότι όσο μεγαλύτερο PSNR έχουμε σαν μετρική στην αποθρομβοποίηση της εικόνας μας κάθε φορά τόσο καλύτερο το αποτέλεσμα αυτής. Αξίζει να σημειωθεί πως έχουμε καλύτερη απόδοση του αλγορίθμου μας για όλους τους τύπους θορύβου που εφαρμόσαμε και για όλες τις εικόνες μας όταν το  $\sigma$  έχει την μικρότερη τιμή, δηλαδή για τις δοκιμές μας  $\sigma=10$ . Ενώ όσο αυξάνουμε την τιμή της τυπικής απόκλισης στον θόρυβο βλέπουμε πως τα αποτελέσματά μας δεν είναι τόσο καλά διότι δημιουργείται μεγαλύτερο θάμπωμα (blurred) κατά την αποθρομβοποίηση της εικόνας

Στη συνέχεια ασχοληθήκαμε με το image inpainting με χρήση πάλι του αλγορίθμου k-SVD. Το image inpainting είναι μία τεχνική για να καλύψουμε χαμένες ή κατεστραμμένες περιοχές από μία εικόνα χρησιμοποιώντας τις διαθέσιμες

πληροφορίες από τη γύρω περιοχή. Ειδικότερα η τεχνική του image inpainting επικεντρώνεται στη διαδικασία ανακατασκευής τμημάτων ή αφαίρεσης τμημάτων μίας εικόνας τα οποία έχουν τροποποιηθεί ή αλλοιωθεί. Η χρήση του k-SVD στην τεχνική του image inpainting έχει ως στόχο την ανάκτηση χαμένης ή κατεστραμμένης πληροφορίας από μία εικόνα με όσο το δυνατόν πιο άρτιο αποτέλεσμα.

Προσαρμόσαμε και εκτελέσαμε τον αλγόριθμο k-SVD και για την τεχνική του image inpainting. Καταλήξαμε ύστερα από πληθώρα δοκιμών, πως όταν θέλουμε να εφαρμόσουμε τον αλγόριθμο για να αφαιρέσουμε για παράδειγμα κάποια γράμματα από μία εικόνα ο αλγόριθμος μας έχει εξίσου ικανοποιητικά αποτελέσματα όπως και στην χρήση του για την αποθορυβοποίηση εικόνων. Παρατηρήσαμε επίσης πως το μέγεθος των γραμμάτων που αφαιρούμε από μία εικόνα επηρεάζει το τελικό αποτέλεσμα της εικόνας μας, ενώ τα αν η εικόνα μας είναι έγχρωμη ή όχι δεν επηρεάζει το αποτέλεσμα του αλγορίθμου μας.

Ύστερα από αρκετή μελέτη και πολλές δοκιμές συμπεράναμε ότι ο αλγόριθμος k-SVD χαρακτηρίζεται από ευελιξία και αποδοτικότητα. Είναι ένας αλγόριθμος με τον οποίο μπορούμε να επιτύχουμε βέλτιστα αποτελέσματα κατά την επεξεργασία και αποθορυβοποίηση μιας εικόνας χρησιμοποιώντας ταυτόχρονα τεχνικές αραιής κωδικοποίησης. Τέλος μέσω αρκετών δοκιμών συμπεράναμε πως ο αλγόριθμος k-SVD μπορεί να χρησιμοποιηθεί τόσο για την αποθορυβοποίηση εικόνας από γνωστό θόρυβο όσο και για την ανάκτηση χαμένων ή κατεστραμμένων τμημάτων εικόνας και να έχει αρκετά καλά αποτελέσματα και στις δύο περιπτώσεις μελέτης.

Ο συνδυασμός του αλγορίθμου K-SVD με τεχνικές μηχανικής μάθησης (deep k-SVD) δύναται να αποτελέσει μια μελλοντική επέκταση της παρούσας εργασίας. Η εκπαίδευση του αλγορίθμου κατά τη διάρκεια εκτέλεσης του εξάγει συγκριτικά καλύτερα αποτελέσματα στον τομέα της αποθορυβοποίησης εικόνων αλλά και ανακατασκευής τους σε σχέση με την απλή εκδοχή του που μελετήθηκε στην παρούσα εργασία.

## Βιβλιογραφία

- [1] Aharon M., Elad M., Bruckstein A., k-SVD An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54 NO. 11, 4311-4322, 2006.
- [2] Chen G., Needekk D., Compressed Sensing and Dictionary Learning, Proceedings of Proceedings of Symposia in Applied Mathematics 73, 2016.
- [3] Candes E., The restricted isometry property and its implications for compressed sensing, Comptes Rendus Mathematique, 346(9), 589 -592, 2008
- [4] Chen S., Donoho D., Saunders M., Atomic decomposition by basis pursuit, SIAM Rev., 43(1), 129-159, 2001.
- [5] Schnass K., A personal introduction to theoretical dictionary learning, Internationale Mathematische Nachrichten Nr. 228, 5-15, 2015.
- [6] Theodoridis S., Machine Learning : A Bayesian and Optimization Perspective, Academic Press, 2015.
- [7] Jonathon Shlens A Tutorial on Principal Component Analysis, Systems Neurobiology Laboratory, Salk Insitute for Biological Studies La Jolla, CA 92037 and Institute for Nonlinear Science, University of California San Diego La Jolla, CA 92092-0402 (10-10-2005, Version 2).
- [8] K. Akhill et al 2019 IOP Conf. Ser.: Mater Sci. Eng 590 012041, Design and Analysis of K-SVD Algorithm for the Noice Removal Using Matlab.
- [9] Marc Lebrun, Arthur Leclair, An Implementation and Detailed Analysis of K-SVD Image Denoising Algorithm, Published in Image Processing On Line on 13-06-2012, ISSN 2105-1232.
- [10] Baker K., Singular Value Decomposition Tutorial 2013, <https://www.ling.ohio-state.edu/.denoising> employing Semi-Parametric Regularization, IEEE Transactions on Signal Processing 19 (6):1465 – 1479, 2010.
- [11] Bouboulis P., Slavakis K., Theodoridis S., Adaptive Kernel-based Image Denoising employing Semi-Parametric Regularization, IEEE Transactions on Signal Processing 19 (6):1465 – 1479, 2010.

- [12] Boyd S., Vandenberghe L., Convex Optimization, Cambridge University Press 2004.
- [13] Elad M., Sparse and Redundant Representations, Springer 2010.
- [14] Devenport M., Du arte M., Eldar Y., Kutyniok G., Introduction to Compressed Sensing Compressed Sensing: Theory and Applications, Cambridge University Press 2012.
- [15] Gallier J., Fundamentals of Linear Algebra and Optimization, Course Notes, 2013, <http://www.cis.upenn.edu/>.
- [16] Mairal J., Elad M., Sapiro G., Sparse learned representations for image Restoration, IASC Yokohama, 2008.
- [17] Mairal J., Elad M., Sapiro G. Sparse representation for Color Image Restoration, IEEE Transactions on Image Processing, 17 (1), 53-69, 2008.
- [18] Shen Y., Han B., Braverman E., Image inpainting using directional tensor product complex tight framelets, Mathematics Subject Classification, 2010.
- [19] Βραχάτης Μ. Ν., Αριθμητική Ανάλυση, Ελληνικά Γράμματα, 2002.
- [20] Otmar Scherzer editor, Handbook of Mathematical Methods on Imaging (second edition), ISBN 978 – 1 – 4939-0789-2, ISBN 978 – 1 – 4939 – 0790 – 8.
- [21] Elad M., Aharon M., Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries, IEEE Transactions on Image Processing 15(12), 3736-3745, 2006.
- [22] Fundamentals of Image Processing hany.farid@dartmouth.edu <http://www.cs.dartmouth.edu/~farid>.
- [23] Image Inpainting Marcelo Bertalmio and Guollermo Sapiro Electrical and Computer Engineering, University of Minnesota, Vicent Caselles and Coloma Ballester Escola Superior Politecnica, Universitat Pompeu Fabra.
- [24] Image inpainting usong image interpolation – An Analysis, P. Jini, Dr K.K. Rajkumar, School of Information Science and Technology, Kannur University Campus, Mangattuparamba, Kannur University Kerala, India.
- [25] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, “Image inpainting” in SIGGRAPH, pp. 417-424,2000.

[26] Free-Form Image Inpainting with Gated Convolution Jiahui Yu, Zhe Lin, Xiaohui Shen, Xin Lu, Thomas Huang, University of Illinois at Urbana-Champaign, Adobe Research, ByteDance AI Lab.

[27] Image Inpainting Based on an Improved K-SVD Counterpart Algorithm, Qingpei Wang, Donghai Bao, Huang Bai, Liping Chang, Sheng Li and Hong Xu College of Information Engineering, Zhejiang University of Technology Hangzhou 310023, Zhejiang, P.R. China clping@zjut.edu.cn.