Πολυτεχνείο Κρήτης

Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών

Μέθοδοι αναγνώρισης γραφής σε μαρμάρινες επιγραφές

Αλεξίου Ελπίδα

Επιτροπή:

καθηγητής Μιχάλης Ζερβάκης (επιβλέπων)

καθηγητής Κωνσταντίνος Μπάλλας

καθηγητής Ευριπίδης Πετράκης

Χανιά Αύγουστος 2003

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου κ. Μιχάλη Ζερβάκη για την πολύτιμη καθοδήγησή και την άψογη συνεργασία του καθώς και στους φίλους μου, προπτυχιακούς και μεταπτυχιακούς για την στήριξη που μου προσέφεραν και για τις πολύτιμες συμβουλές τους.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ

Στην εργασία αυτή επιχειρείται η μελέτη εικόνων που έχουν ανακτηθεί από μαρμάρινες επιγραφές με μεθόδους επεξεργασίας εικόνας. Πιο συγκεκριμένα υλοποιείται ένα σύστημα που έχει στόχο την αναγνώριση των χαρακτήρων της επιγραφής καθώς και την βελτίωση του thresh□ld για την μετατροπή της εικόνας σε ασπρόμαυρη.

Ο βασικός αλγόριθμος που χρησιμοποιείται για την αναγνώριση βασίζεται στην μέθοδο των ροπών. Οι ροπές έχουν χρησιμοποιηθεί σε έναν αριθμό εφαρμογών προκειμένου να χρησιμοποιηθούν ως αμετάβλητα χαρακτηριστικά σε προβλήματα αναγνώρισης διδιάστατων προτύπων. Για να φτάσουμε στην υλοποίηση του αλγορίθμου Ge□metric M□ments χρειάστηκε η υλοποίηση και κάποιων άλλων χρήσιμων αλγορίθμων όπως του Minimum Spanning Tree και της εύρεσης περιγράμματος που περιγράφονται παρακάτω πιο αναλυτικά.

Η βελτίωση του thresh□ld στηρίχτηκε στις ιδιότητες του ιστογράμματος και στην πληροφορία που μας παρέχει αυτό για το περιεχόμενο της εικόνας. Επίσης, βασιζόμενοι στο Thresh□ld που μας δίνει ο αλγόριθμος του □tsu καταφέραμε να βελτιώσουμε την ποιότητα της ασπρόμαυρης εικόνας με m□rph□l□gical □perati□ns που εξαλείφουν τον θόρυβο και πέρνουμε ως αποτέλεσμα πιο ευδιάκριτο τον χαρακτήρα της επιγραφής έτσι ώστε να έχουμε καλύτερα αποτελέσματα στην αναγνώριση.

1. Εισαγωγή

1.1 Καθορισμός του προβλήματος

Η αρχαιολογική και ιστορική σημασία των αρχαίων επιγραφών είναι ανεκτίμητη αφού με αυτόν τον τρόπο έχει μείνει αποτυπωμένος στο χρόνο ο γραπτός λόγος των προγόνων μας. Συνεχής είναι η προσπάθεια μελέτης αλλά και συντήρησης αυτής της πολιτιστικής κληρονομιάς από αρχαιολόγους και συντηρητές αρχαιοτήτων. Δεν είναι λίγα τα προβλήματα που αντιμετωπίζουν στην προσπάθειά τους αυτή αν λάβουμε υπόψη μας ότι οι περισσότερες από αυτές τις επιγραφές είναι χαραγμένες σε μάρμαρο ή πέτρα, υλικά που η αρχική τους μορφή με το πέρασμα του χρόνου αλλοιώθηκε. Η έκθεση τους σε εξωγενείς παράγοντες, όπως νερό και αέρας, συντέλεσε στη διάβρωση και στην εν μέρει καταστροφή της επιφάνειάς τους.

Στην εργασία αυτή γίνεται μία προσπάθεια μελέτης εικόνων μαρμάρινων επιγραφών με μεθόδους image pr□cessing. Πιο συγκεκριμένα υλοποιείται ένα σύστημα αναγνώρισης γραφής σε μαρμάρινες επιγραφές. Έτσι αρχικό μας μέλημα ήταν η εξαγωγή του χαρακτήρα από την εικόνα ή οι περιοχές που περιέχουν κομμάτια κάποιου γράμματος εφόσον αυτό ήταν σπασμένο και η απομόνωση των περιοχών του θορύβου.

Τα τελευταία χρόνια έχει γίνει αρκετή δουλειά για την αξιολόγηση των συστημάτων ανάκτησης πληροφοριών για αλφανουμερικά δεδομένα. Τέτοια συστήματα χρησιμοποιούνται για σύγκριση εικόνων με σκοπό και την αναγνώριση σχημάτων και χαρακτήρων.

Ένας σημαντικός τομέας της έρευνας σε αυτά τα αναδυόμενα συστήματα είναι ο αυτόματος χαρακτηρισμός του περιεχομένου εικόνας και ανάκτηση των εικόνων βασισμένων στην ομοιότητα του περιεχομένου αυτού. Στα συστήματα πληροφοριών εικόνας ένα κοινό κριτήριο για την ανάκτηση στοιχείων είναι «ποιο αντικείμενο (σχήμα ή χαρακτήρας) από μια βάση δεδομένων ταιριάζει ή είναι πολύ παρόμοιο με ένα δεδομένο σχήμα ή χαρακτήρα?» Αυτός ο τύπος ανάκτησης λέγεται **Shape Similarity Based Retrieval**.

Στην εργασία αυτή βασιστήκαμε σε μια τέτοια μέθοδο για την ανάκτηση πληροφορίας από τις εικόνες με σκοπό την σύγκριση και αναγνώριση των χαρακτήρων που περιέχουν.

Ένας επιπλέον στόχος της εργασίας είναι η εκτίμηση ενός καλού thresh□ld για την μετατροπή μιας εικόνας σε ασπρόμαυρη, με μεθόδους ψηφιακής επεξεργασίας εικόνας. Έτσι η εργασία εστιάζεται στην επεξεργασία των δεδομένων παρά σε μεθόδους πιστής αποτύπωσης της εικόνας της επιγραφής. Για το σκοπό αυτό χρησιμοποιήθηκε μία απλή κάμερα (ψηφιακή) με flash, χωρίς να δίνεται ιδιαίτερη σημασία στην τοποθέτηση της κάμερας. Ένα τέτοιο σύστημα έχει πολύ χαμηλό κόστος, μπορεί να χρησιμοποιηθεί από τον οποιονδήποτε και σε ένα μεγάλο εύρος εφαρμογών, πέραν αυτής που εξετάζουμε.

Οι δυσκολίες που αντιμετωπίσαμε κατά την επεξεργασία της εικόνας οφείλονται κυρίως στην ποιότητά της.

Για να οριστεί καλύτερα ο όρος «ποιότητα» θα πρέπει να λάβουμε υπόψη μας ότι τα γράμματα χαρακτηρίζονται από μία ποικιλία ως προς τη μορφή τους. Συναντάμε λοιπόν χαρακτήρες με πολύ λεπτή δομή, άλλους περισσότερο ευκρινείς με πιο ισχυρή δομή, «σπασμένους» χαρακτήρες και τέλος χαρακτήρες με έντονη εκτεταμένη δομή. Η μορφή με την οποία παρουσιάζεται το κάθε γράμμα οφείλεται στην ανομοιομορφία του μαρμάρου στο οποίο είναι χαραγμένα τα γράμματα. Στη συνολική επιφάνεια του μαρμάρου συναντάμε κάποιες σχετικά ομαλές περιοχές, περιοχές με περιοδική (πορώδη) διάβρωση, άλλες με γραμμική διάβρωση με τη μορφή επιμηκών δομών καθώς επίσης σπασίματα και πολύ έντονες τυχαίες γραμμώσεις («χαρακιές»). Γενικά, μία υποπεριοχή θεωρείται ότι έχει *αποδεκτή ποιότητα* όταν περιέχει χαμηλό ποσοστό θορύβου και μικρό ποσοστό διάβρωσης και όταν το γράμμα που τυχόν υπάρχει σε αυτήν την περιοχή είναι ευκρινές.

Παράγοντες που δυσχεραίνουν επιπλέον την προσπάθεια εκτίμησης είναι η ανομοιομορφία του φωτισμού της εικόνας καθώς και η κακή ανάλυσή της, στοιχεία που αλλοιώνουν κατά κάποιο τρόπο την πληροφορία που υπάρχει στην εικόνα και επηρεάζουν αρνητικά την επεξεργασία της. Και τα δύο αυτά προβλήματα προκύπτουν από το σύστημα ανάκτησης της εικόνας που ίσως αν ήταν καλύτερο (καλύτερη ψηφιακή camera) τα προβλήματα αυτά να μην ήταν τόσο έντονα.

Η εργασία αυτή έχει την ακόλουθη δομή: Στην *Παράγραφο 1.2* γίνεται μία προεργασία στα πλαίσια της οποίας τμηματοποιείται η εικόνα σε υποπεριοχές στις οποίες εφαρμόζονται όλοι οι μέθοδοι που θα αναπτύξουμε στη συνέχεια και γίνεται μία προσπάθεια να εξομαλυνθούν ως ένα βαθμό τόσο τα προβλήματα του συστήματος ανάκτησης όσο και τα προβλήματα του ανομοιόμορφου backgr□und του μαρμάρου.

Στο 2^{ο} κεφάλαιο αναλύεται ένα από τα στάδια της επεξεργασίας , που είναι η μετατροπή της εικόνας σε ασπρόμαυρη με βάση κάποιο Thresh□ld. Παρουσιάζονται μέθοδοι καθαρισμού της εικόνας από τον θόρυβο καθώς και τρόπος βελτίωσης της τιμής του Thresh□ld που μας δίνει ο αλγόριθμος του □tsu.

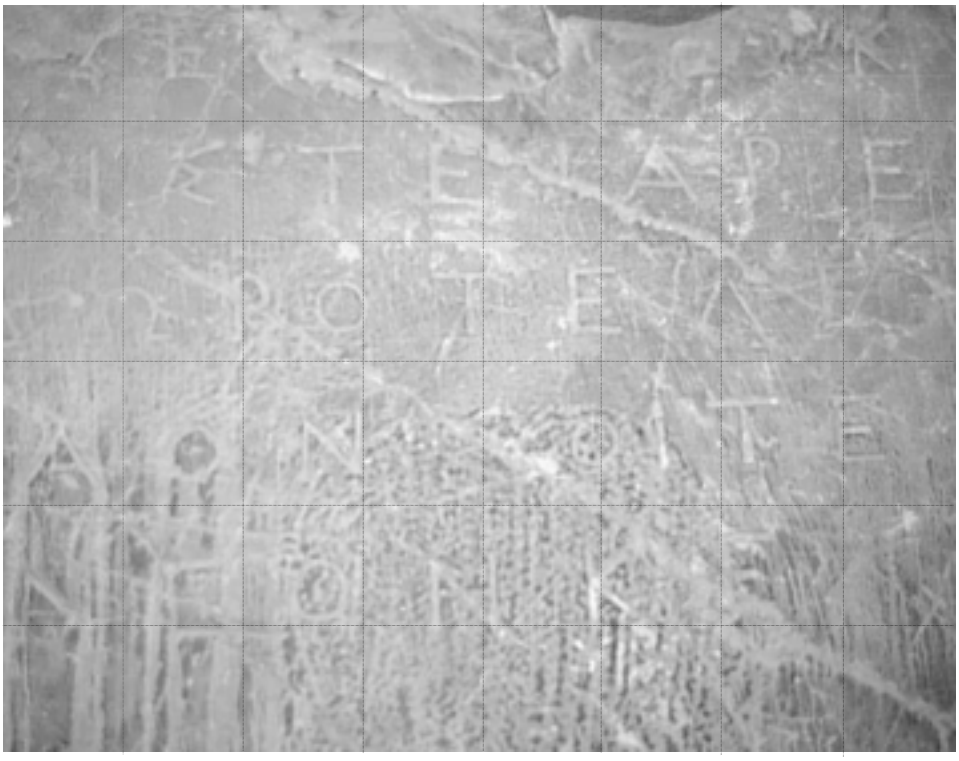
Στο 3^{ο} κεφάλαιο υπάρχει μια περιγραφή του αλγορίθμου εξαγωγής περιγράμματος καθώς και παραδείγματα εφαρμογής του όπως η εύρεση ελάχιστης απόστασης αντικειμένων με βάση τα σημεία των περιγραμμάτων τους.

Στο 3^{ο} κεφάλαιο επίσης, παρουσιάζεται ο αλγόριθμος Minimum Spanning Tree που χρησιμοποιήθηκε για τον εντοπισμό περιοχών γειτονικών κομματιών. Ο αλγόριθμος αυτός χρησιμοποιεί τις ελάχιστες αποστάσεις που βρέθηκαν με βάση τον αλγόριθμο του περιγράμματος.

Στο 4^{ο} και τελευταίο κεφάλαιο βλέπουμε τον αλγόριθμο Ge□metric M□ments που χρησιμοποιείται για την αναγνώριση των χαρακτήρων. Ο αλγόριθμος δέχεται σαν είσοδο τις εικόνες με τις περιοχές που βρέθηκαν από τον αλγόριθμο Minimum Spanning Tree που είναι αυτές που πιθανότατα θα περιέχουν κομμάτια κάποιου γράμματος και μας δίνει σαν έξοδο το κοντινότερο χαρακτήρα .

1.2 Pre-pr□cessing

□ αρχική εικόνα (*Εικόνα 1*) χωρίζεται σε πολλά κομμάτια (segments) με σκοπό να απομονωθούν καλύτερα οι διάφορες περιοχές και η επεξεργασία τους να γίνει με μεγαλύτερη λεπτομέρεια. Η εικόνα διαιρείται με τέτοιον τρόπο έτσι ώστε στα διάφορα segments να περιέχονται κατά το δυνατόν ολόκληροι χαρακτήρες και το κάθε segment να περιέχει της ίδιας περίπου μορφής backgr□und. Για να επιτευχθεί αυτό η εικόνα διαιρείται σε 48 κομμάτια διαστάσεων 80x80. Άλλοι μέθοδοι τμηματοποίησης της εικόνας συζητούνται στις μελλοντικές επεκτάσεις.



Εικόνα 1: Μία από τις εικόνες της μαρμάρινης επιγραφής στην οποία βασίστηκε η εργασία. Στην εικόνα αυτή φαίνεται και πως ακριβώς έχει γίνει το segmentation.

Στην συνέχεια προχωράμε στην επεξεργασία κάθε segment χωριστά. Πρέπει να σημειωθεί ότι δεν χρησιμοποιούμε φίλτρα εξομάλυνσης για να μην αλλοιώσουμε τα χαρακτηριστικά των ακμών των γραμμάτων που μας είναι χρήσιμα για την περεταίρω επεξεργασία της εικόνας.

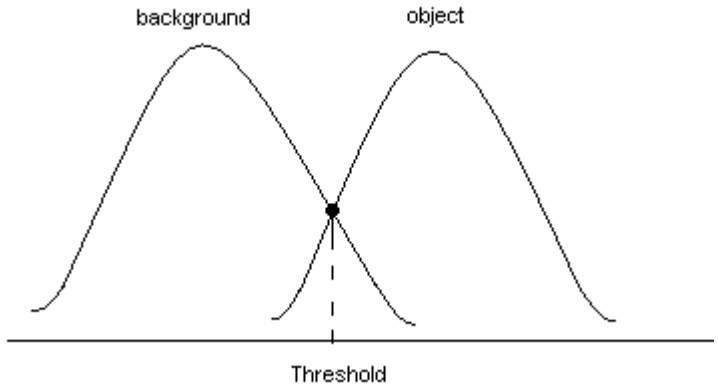
Επόμενο, λοιπόν, βήμα μας είναι η μετατροπή των segment σε ασπρόμαυρες (binary) εικόνες me βάση την τιμή του Threshold της grayscale εικόνας μας και ο καθαρισμός της ασπρόμαυρης πια εικόνας μας από τον θόρυβο. τρόπος παρουσιάζεται αναλυτικά παρακάτω. Ο λόγος που μετατρέπουμε τα Segments σε ασπρόμαυρες εικόνες είναι ότι υπάρχει μεγάλη γκάμα εργαλείων για Image processing σε ασπρόμαυρες εικόνες.

2. □hresh□ld

2.1 Μετατροπή μιας εικόνας σε ασπρόμαυρη με το thresh□ld του □tsu.

Για την μετατροπή μιας εικόνας σε ασπρόμαυρη χρειάζεται κάποιο Thresh□ld που παίρνει τιμές στο διάστημα [0,1] και λειτουργεί ως εξής : Η ασπόμαυρη εικόνα που προκύπτει έχει μηδενικές τιμές, δηλαδή μαύρο για εκείνα τα pixels της αρχικής εικόνας με φωτεινότητα μικρότερη του Thresh□ld και άσσους, δηλαδή άσπρο για όλα τα υπόλοιπα pixels.

Για τον υπολογισμό αυτού του Thresh□ld συνήθως χρησιμοποιείται η μέθοδος του □tsu, η οποία όμως στην περίπτωσή μας δεν έχει τα επιθυμητά αποτελέσματα. Στην μέθοδο αυτή κρατούνται στατιστικά στοιχεία για δύο κλάσσεις τιμών έντασης, μία του αντικειμένου και μία του φόντου. Τις δύο αυτές κλάσσεις ο □tsu τις περιγράφει με δυο ξεχωριστές Gaussian κατανομές σε διαφορετικές τιμές έντασης. Το σημείο που τέμνονται ο □tsu επιλέγει σαν ιδανικό Thresh□ld όπως φαίνεται παρακάτω.

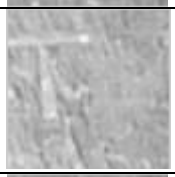
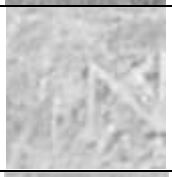
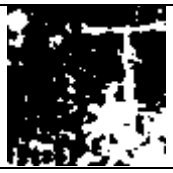


□ λόγος για τον οποίο το Thresh□ld του □tsu δεν μας κάνει είναι ότι το αντικείμενο που θέλουμε να εξάγουμε από την εικόνα μας δεν έχει και τόσο διαφορετικά χαρακτηριστικά από το φόντο και έτσι δεν μπορεί να περιγραφεί σαν μια ξεχωριστή Gaussian κατανομή.

Αν η επιλογή του Thresh□ld γίνει manual καταλήγουμε σε κάποια ιδανική τιμή που συνήθως μας δίνει τα επιθυμητά αποτελέσματα. Αυτό όμως δεν είναι πρακτικό και λειτουργικό. Έτσι καταφέραμε να βελτιώσουμε τα αποτελέσματα του segmentati□n που

προκύπτει με το Thresh□ld του □tsu με την χρήση M□rph□l□gical □perati□ns πάνω στην εικόνα όπως θα δούμε σε επόμενη παράγραφο.

Παρακάτω φαίνονται μερικά παραδείγματα που αποδεικνύουν ότι το Thresh□ld του □tsu δεν μας βοηθάει για την περεταίρω επεξεργασία.

Αρχική εικόνα	**Ασπρόμαυρη με το thresh□ld του □tsu**	**Ασπρόμαυρη με ιδανικό □hresh□ld**
		
		
		
		
		

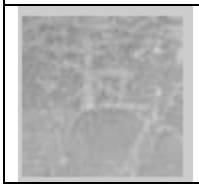
2.2 Gaussian and exp□nential fitting

Κάνοντας μια προσπάθεια να βελτιώσουμε την τιμή του thresh□ld βασιστήκαμε στην πληροφορία που μας παρέχει το ιστόγραμμα μιας εικόνας. Αν υποθέσουμε ότι η αρχική μας εικόνα έχει L διακριτά επίπεδα του γκρι (συνήθως από 0 μέχρι 255) και ότι n_k, k=0,1,…,L-1 είναι ο αριθμός των pixels που έχουν φωτεινότητα k, τότε, το ιστόγραμμα δίνεται από την σχέση :

$$\mathbf{p_f(f_k)= n_k / n}\ ,\ k=0,1,\ldots,L-1$$

όπου n είναι ο συνολικός αριθμός των pixels της εικόνας.

Στην περίπτωσή μας, όπως και στον Otsu ο θόρυβος υπόκειται σε Gaussian μοντελοποίηση. Τα γράμματα όμως λόγω του ότι είναι μικρού σχετικά μεγέθους δεν εμφανίζουν έντονη κατανομή. Έρχονται σαν outliers μιας κανονικής κατανομής που περιγράφει το background. Έτσι εμείς υποθέσαμε ότι αυτήν την κατανομή περιγράφει μια εκθετική συνάρτηση που ξεκινά από την μέση τιμή της Gaussian κατανομής. Αρχικά λοιπόν κανονικοποιήσαμε το ιστόγραμμα κάνοντας fit μια Gaussian κατανομή. Έτσι εξομαλύναμε λίγο τις απότομες διακυμάνσεις του ιστογράμματος. Στην συνέχεια από την μέση τιμή της Gaussian κατανομής κάναμε fit μια εκθετική συνάρτηση. Το σημείο που η εκθετική τέμνει την κατανομή Gauss επιλέγουμε σαν Threshold αφού τις περισσότερες φορές από αυτό το σημείο και μετά θα υπάρχει η περιοχή που περιλαμβάνει τα σημεία του γράμματος. Στις εικόνες όπου το γράμμα μας είναι ευδιάκριτο και ξεχωρίζει αρκετά από το background η μέθοδός μας είχε καλύτερα αποτελέσματα από τον Otsu. Παρακάτω φαίνονται γραφικά τα αποτελέσματα όπου με μωβ στίγματα απεικονίζεται το ιστόγραμμα, με κόκκινη γραμμή η Gaussian κατανομή και με μπλε γραμμή η εκθετική :

Image	**Otsu's threshold**	**Gaussian & exponential threshold**	**Ideal threshold**
	**0.66**	**0.67**	**0.68**

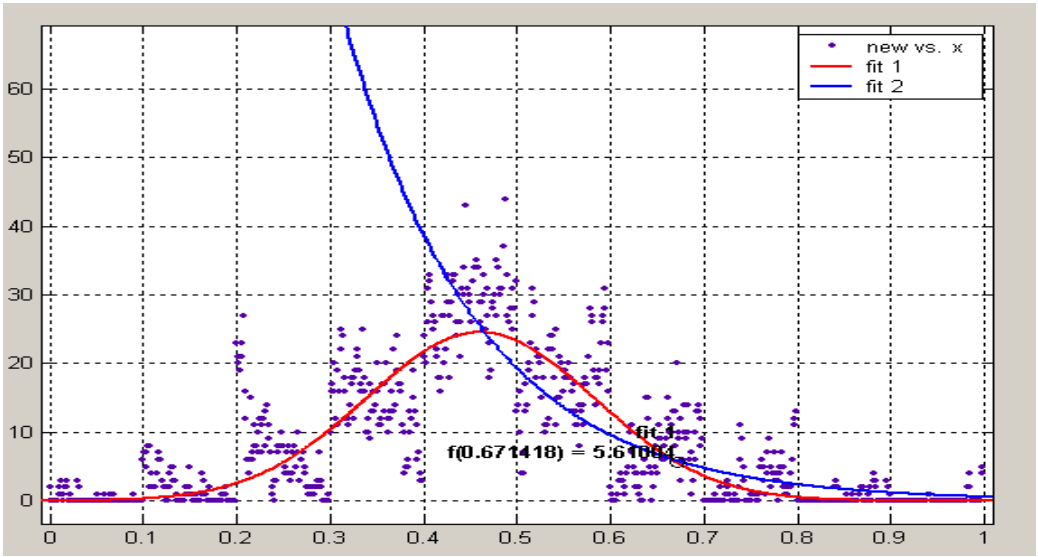
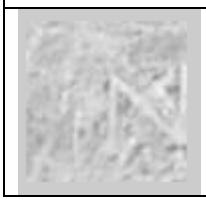


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.78**	**0.80**	**0.82**

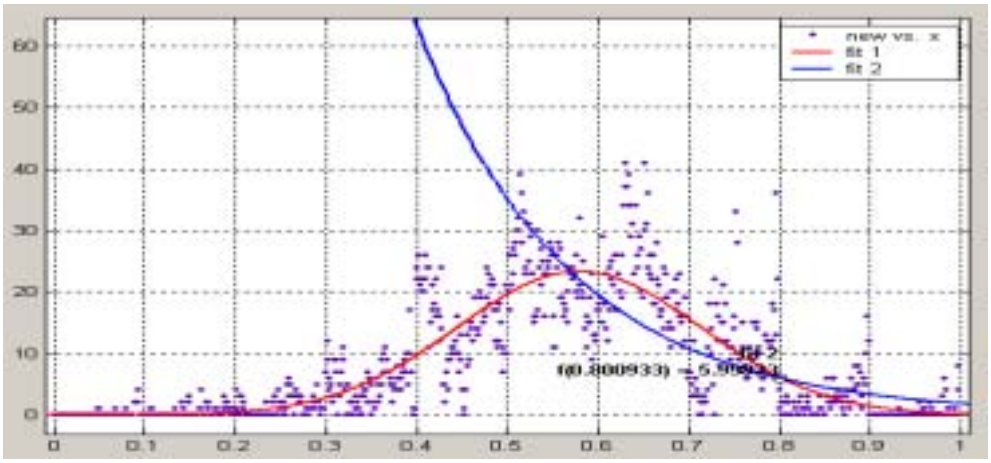
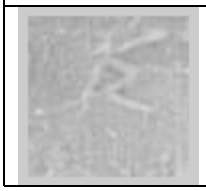


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.72**	**0.73**	**0.74**

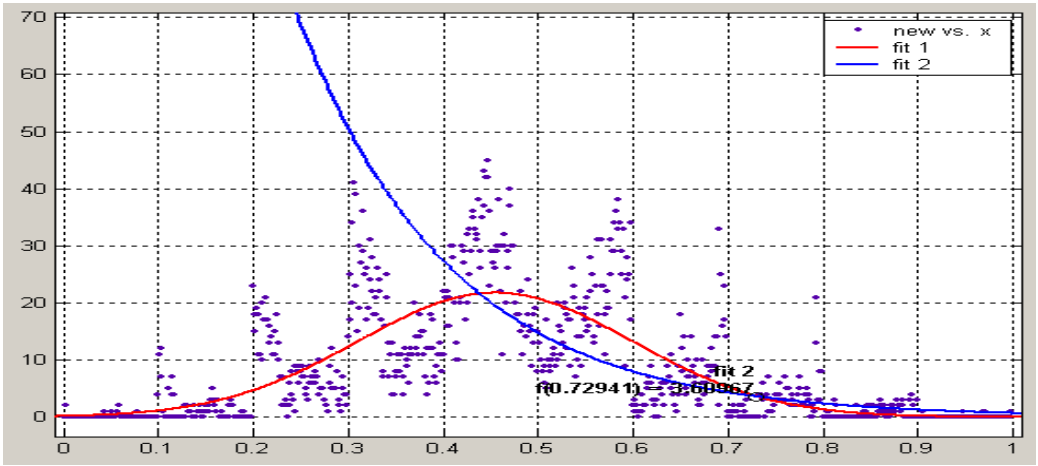
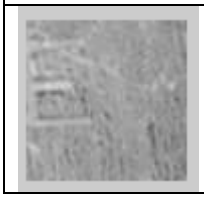


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.62**	**0.64**	**0.68**

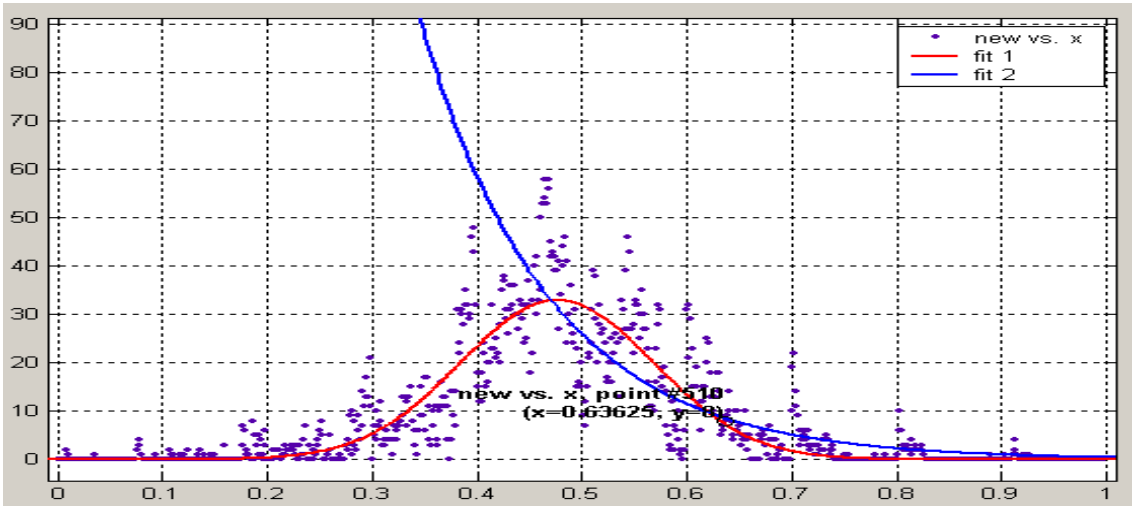
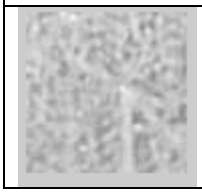


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.73**	**0.75**	**0.76**

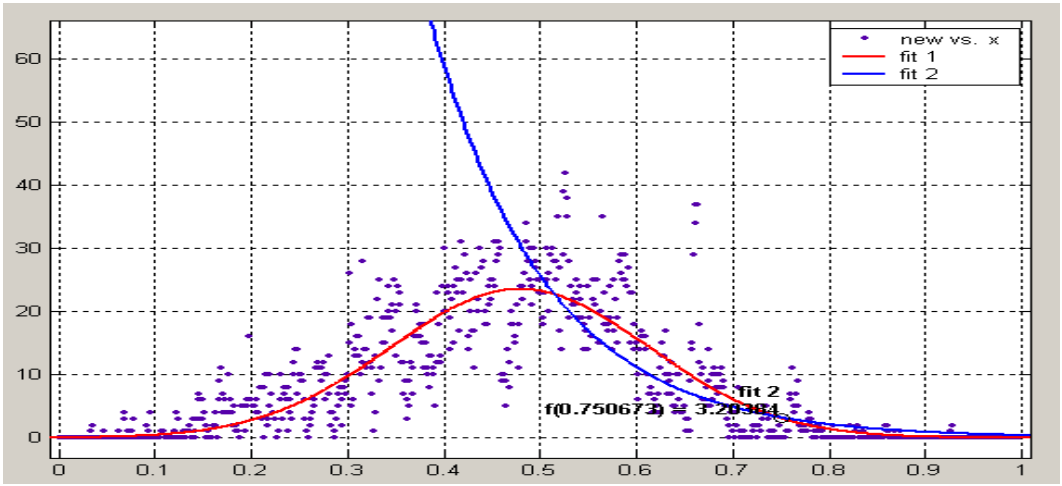
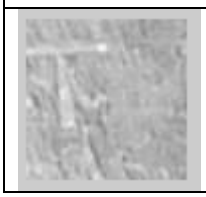


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.71**	**0.79**	**0.77**

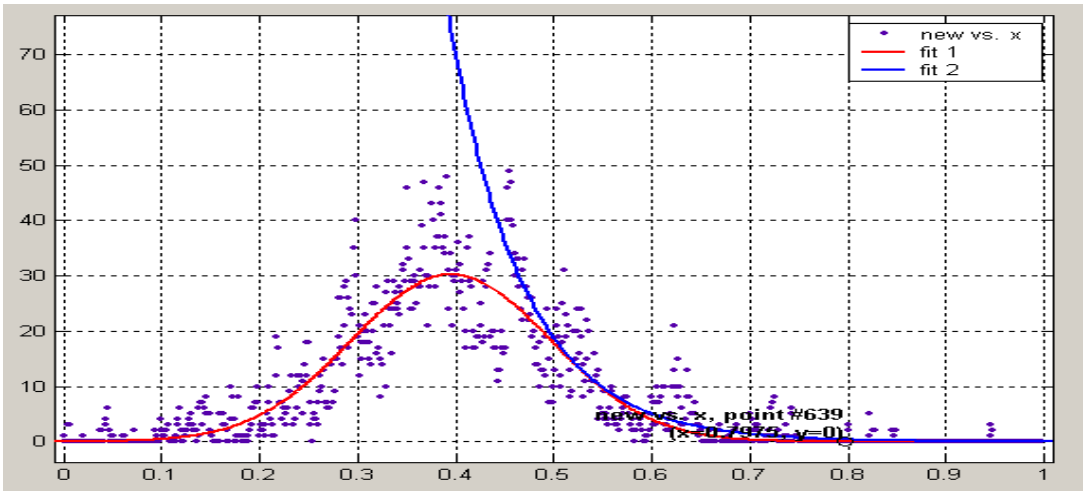
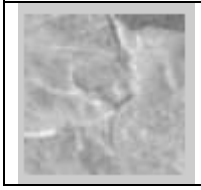


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.68**	**0.7**	**0.75**

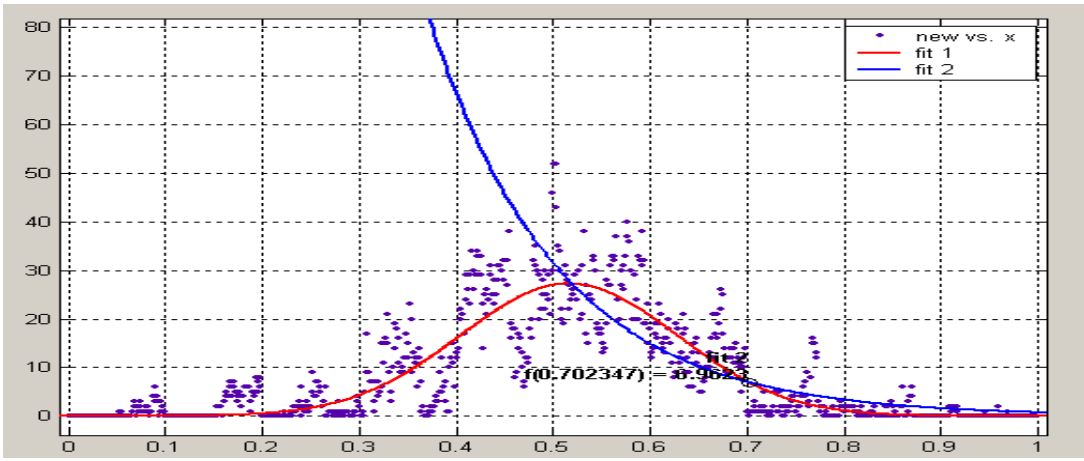
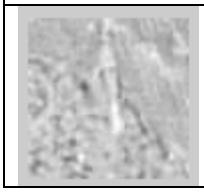
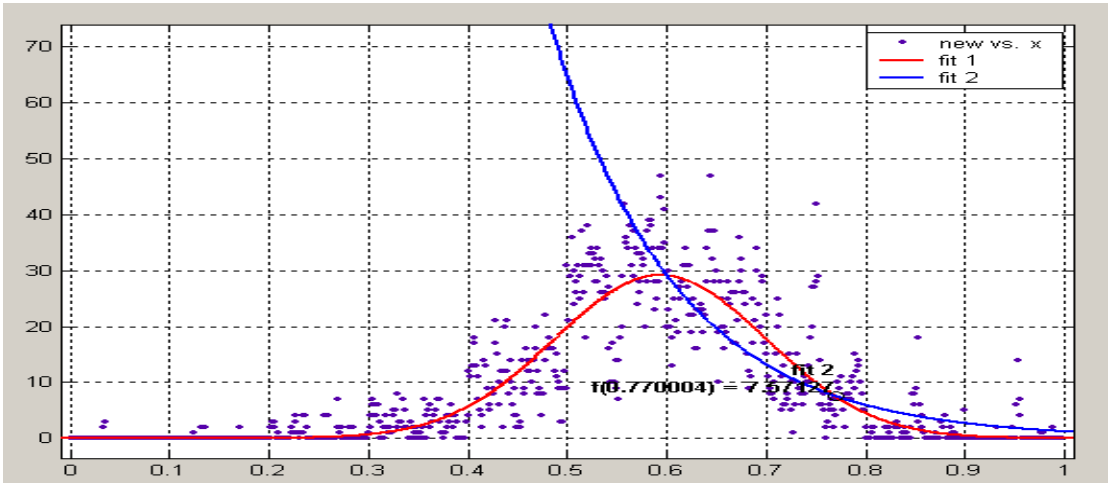


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.74**	**0.77**	**0.82**



2.3 Πίνακας αποτελεσμάτων

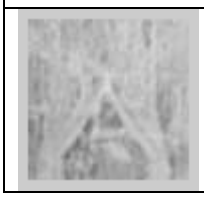
Στον παρακάτω πίνακα φαίνονται συνολικά για όλες τις εικόνες οι τιμές των thresh□lds του □tsu, του fitting του ιστογράμματος με Gaussian και εκθετική κατανομή και η ιδανική τιμή.

Με πράσινο χρώμα φαίνονται οι περιπτώσεις όπου το Thresh□ld που προκύπτει από το Gaussian & exp□nential fitting είναι πιο κοντά στο ιδανικό από το Thresh□ld του □tsu. Με ♦ συμβολίζονται οι εικόνες που περιέχουν γράμμα ενώ με ⊗ οι εικόνες που έχουν απορριφθεί από προηγούμενη εργασία. Στις εικόνες που δεν περιέχουν γράμμα , είναι λογικό να αστοχούμε αφού μοντελοποιούμε τον χαρακτήρα που στην περίπτωση αυτή δεν υπάρχει με μια εκθετική κατανομή.

Image number	□tsu	Gaussian/□xp□nential fit	□ anual
1 ⊗	0.61	0.86	0.65
2 ♦	0,66	0.67	0.68
3⊗	0.73	0.71	0.82
4⊗	0.76	0.64	0.78
5♦	0.68	0.7	0.75

6⊗	0.58	0.72	0.72
7⊗	0.65	0.69	0.70
8♦	0.55	0.67	0.59
9♦	0.66	0.55	0.72
10 ♦	0.72	0.73	0.74
11♦	0.78	0.59	0.79
12♦	0.81	0.61	0.81
13⊗	0.77	0.70	0,81
14♦	0.74	0.69	0.76
15♦	0.71	0.76	0.75
16♦	0.62	0.59	0.66
17⊗	0.70	0.88	0.70
18♦	0.76	0.74	0.76
19♦	0.78	0.70	0.81
20♦	0.79	0.61	0.79
21♦	0.80	0.68	0.82
22	0.74	0.56	0.76
23⊗	0.70	0.58	0.76
24	0.63	0.54	0.66
25♦	0.68	0.95	0.77
26♦	0.72	0.95	0.78
27♦	0.78	0.80	0.82
28⊗	0.76	0.80	0.82
29	0.77	0.80	0.82
30♦	0.74	0.77	0.82
31♦	0,71	0,75	0.77
32♦	0,62	0,64	0.68
33♦	0.63	0.83	0.75
34	0.69	0.89	0.76
35♦	0.75	0.83	0.78
36♦	0.73	0.75	0.76
37⊗	0.72	0.79	0.76
38	0.71	0.78	0.77
39♦	0.65	0.80	0.72
40⊗	0.56	0.72	0.65
41⊗	0.57	0.90	0.65
42⊗	0.63	0.90	0.69
43⊗	0.68	0.80	0.75
44⊗	0.69	0.93	0.76
45⊗	0.67	0.73	0.76
46⊗	0.65	0.79	0.76
47⊗	0.59	0.71	0.65
48⊗	0.47	0.82	0.50

Ενδιαφέρον παρουσιάζουν περιπτώσεις όπου η μέθοδός μας αποκλίνει πολύ όπως οι παρακάτω περιπτώσεις :

Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.68**	**0.95**	**0.77**

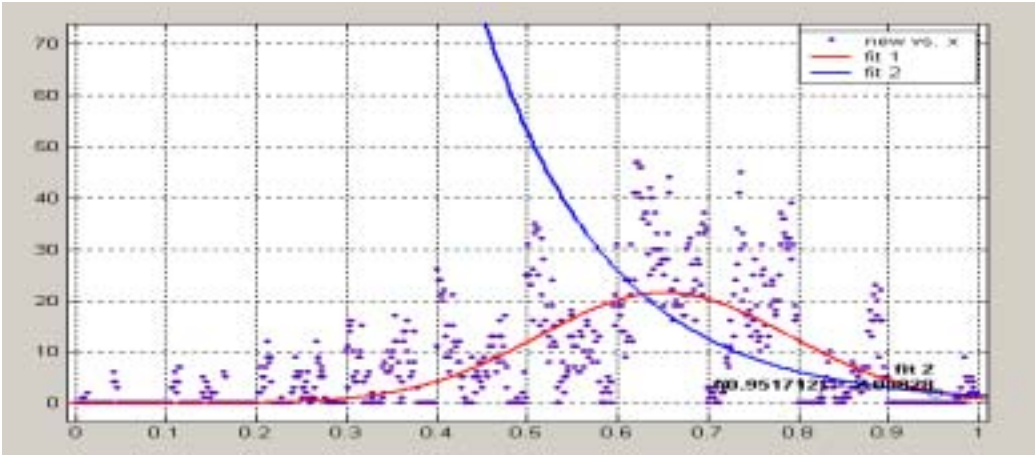
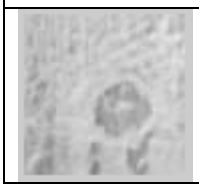


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.72**	**0.96**	**0.78**

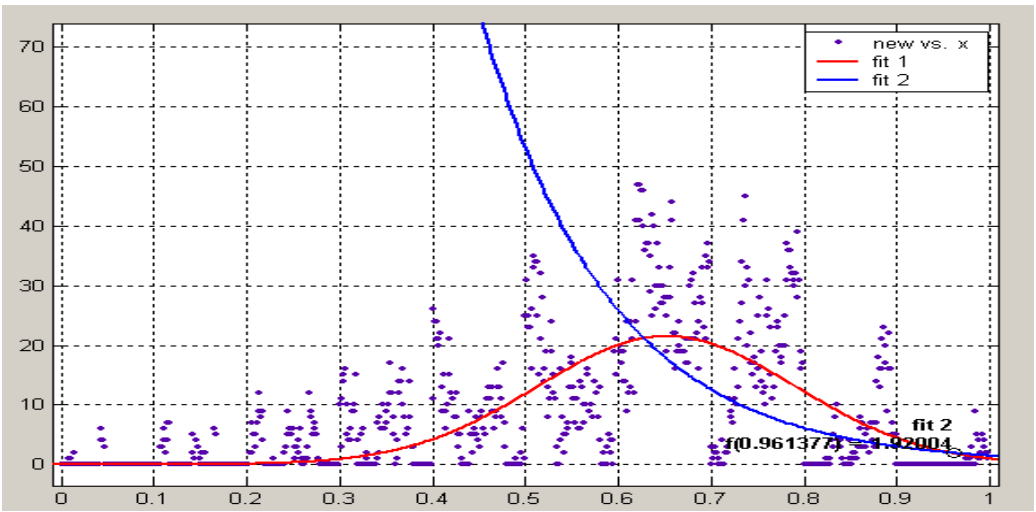
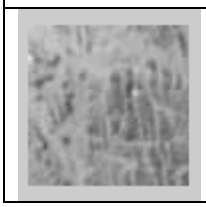


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.59**	**0.71**	**0.65**

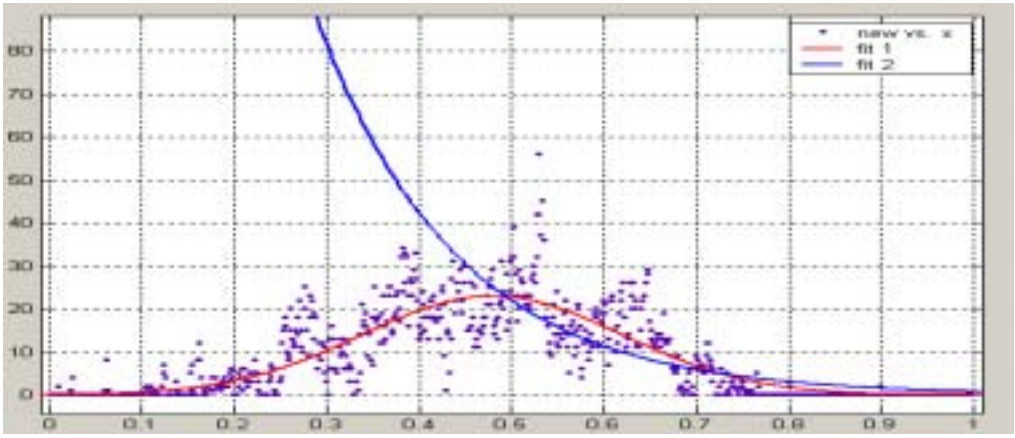
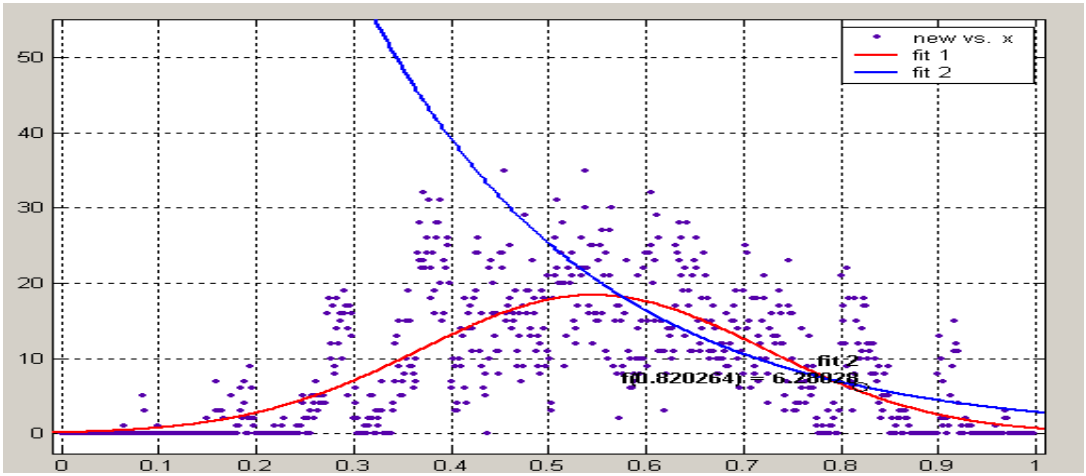


Image	□tsu's thresh□ld	Gaussian & □xp□nential thresh□ld	Ideal thresh□ld
	**0.47**	**0.82**	**0.50**



Αυτές οι περιπτώσεις είναι εικόνες που περιέχουν μόνο θόρυβο ή που το γράμμα είναι ενωμένο με αρκετό θόρυβο και δεν είναι ευδιάκριτο . Έτσι δεν μπορεί να μοντελοποιηθεί με μια εκθετική κατανομή και για αυτό η μέθοδος αυτή αποτυγχάνει.

2.4 □ □rph□l□gical □perati□ns

2.4.1 □ □rph□l□gical □perati□ns στο thresh□ld του Otsu

□πειδή όπως είναι λογικό δεν γίνεται να χρησιμοποιήσουμε την μέθοδο όπου βρίσκουμε manual το κατάλληλο Thresh□ld χρειάστηκε να φτιάξουμε έναν αυτόματο τρόπο ο οποίος, μας βελτιώνει τα αποτελέσματα που πέρνουμε από τον □tsu. Έτσι στις ασπρόμαυρες εικόνες που προκύπτουν από την μετατροπή των αρχικών με το thresh□ld του □tsu εφαρμόσαμε κάποιες τεχνικές που βελτιώνουν την μορφολογία τους. Πιο συγκεκριμένα χρησιμοποιήσαμε:

- Την μέθοδο καθαρισμού **(clean)** η οποία απομακρύνει απομονωμένα pixels δηλαδή άσσους (άσπρα) που περιβάλλονται από μηδενικά (μαύρα) όπως το κεντρικό pixel του παραδείγματος :

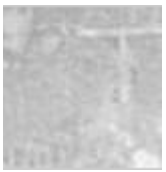
0	**0**	**0**
0	**1**	**0**
0	**0**	**0**

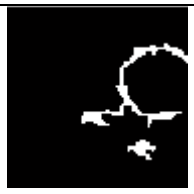
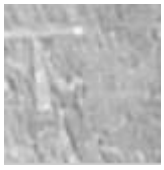
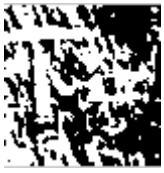
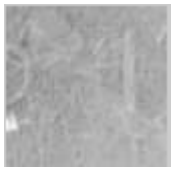
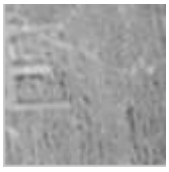
- Την μέθοδο «γεμίσματος» **(fill)** η οποία γεμίζει απομονωμένα εσωτερικά pixels δηλαδή μηδενικά που περιβάλλονται από άσσους όπως το κεντρικό pixel του παραδείγματος:

1	**1**	**1**
1	**0**	**1**
1	**1**	**1**

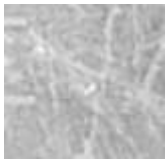
- Την μέθοδο **er□si□n** η οποία απομακρύνει μικρές δομές που είναι προσκολλημένες πάνω στα αντικείμενα. Στην περίπτωσή μας αυτές οι δομές είναι ανεπιθύμητος θόρυβος.
- Τέλος χωρίσαμε κάθε εικόνα με ένα grid 3x3, δηλαδή σε ενιά κομμάτια, και καθαρίσαμε κάθε μία από αυτές τις περιοχές όπου η περιεκτικότητά της σε pixels ξεπερνούσε κάποιο thresh□ld που επειλέξαμε εμπειρικά.

Έτσι καταφέραμε να βελτιώσουμε σημαντικά και με αυτόματο τρόπο τις εικόνες μας όπως φαίνεται σε μερικά παραδείγματα στον παρακάτω πίνακα:

Αρχική Εικόνα	**Ασπρόμαυρη με το □hresh□ld του □tsu**	**Τελική εικόνα**
		
		
		
		

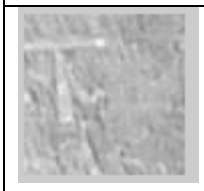
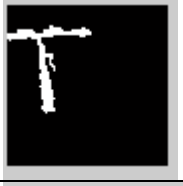
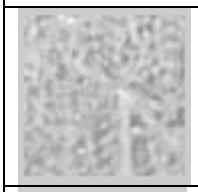
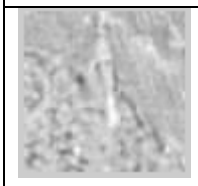
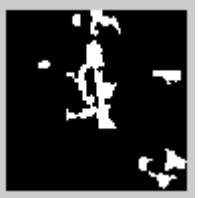
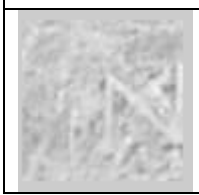
		
		
		
		
		
		
		

Υπάρχουν όμως και περιπτώσεις που οι τεχνικές για την βελτίωση της μορφολογίας των εικόνων μας έχουν σαν αποτέλεσμα να χάνουμε και το γράμμα όπως φαίνεται παρακάτω:

Αρχική Εικόνα	Ασπρόμαυρη με το Thresh□ld του □tsu	Τελική εικόνα
		
		
		

2.4.2 □ □rph□l□gical □perati□ns στο thresh□ld του Fit

Εφαρμόσαμε m□rph□l□gical □perati□ns στις ασπρόμαυρες εικόνες που περιέχουν γράμμα και των οποίων βελτιώσαμε το Thresh□ld με την μέθοδο Gaussian & □xp□nential Fit. Οι μέθοδοι αυτές ήταν παρόμοιες με τις παραπάνω αλλά με μικρές διαφορές αφού οι εικόνες μας τώρα είναι αρκετά διαφορετικές και απαιτούν άλλη επεξεργασία. Μερικά παραδείγματα φαίνονται στον παρακάτω πίνακα:

Αρχική εικόνα	Τελική εικόνα
	
	
	
	
	
	
	
	

2.4.3 □πίδραση του thresh□ld

Από τα μέχρι τώρα αποτελέσματα παρατηρούμε ότι το thresh□ld του □tsu βρίσκεται σε χαμηλότερα επίπεδα από το επιθυμητό και συνεπώς και από το δικό μας βελτιωμένο Thresh□ld που προκύπτει με την μέθοδο του fit. Έτσι εφαρμόζοντας Multiple thresh□ld □perati□n αυξάνοντας την τιμή του thresh□ld που προκύπτει από το fit, αλλά και από τον □tsu, καταφέρνουμε να βελτιώσουμε το segmentati□n σπάζοντας μεν την εικόνα μας σε περισσότερα κομμάτια αλλά διατηρώντας περιοχές του γράμματος αρκετά κοντά ενωμένες απαλλαγμένες όμως από την έντονη παρουσία θορύβου. Παραδείγματα αυτής της διαδικασίας αναφέρονται παρακάτω αφού αυτή εφαρμόστηκε στον αλγόριθμο Ge□metric M□ments για το classificati□n.

3.Αλγοριθμοι εξαγωγής περιγράμματος και □ inimum Spanning □ree

3.1 Υλοποίηση αλγορίθμου εξαγωγής περιγράμματος

Η μέθοδος που υλοποιήθηκε για να βρεθεί το περίγραμμα βασίζεται στον αλγόριθμο των □amada και □asuike. Στην μέθοδο αυτή χρησιμοποιείται μια μάσκα ψαξίματος για τον εντοπισμό του περιγράμματος pixel-pixel. Η μάσκα είναι διαστάσεων (2M+1) x (2N+1) και τα στοιχεία της α_{ij} υπολογίζονται με τις παρακάτω εξισώσεις :

$$\delta\theta_M = \tan^{-1}(N-1 / \square) - \tan^{-1}(N-2 / \square -1)$$

$$\delta\theta_N = \tan^{-1}(\square -1 / N) - \tan^{-1}(\square -2 / N-1)$$

$$\alpha_{ij} = \text{atan2}(j, i) + \delta\theta_M \, \delta\theta_N \left(1 - \left(\sqrt{i^2 + j^2}\right) / \left(\sqrt{\square^2 + N^2}\right)\right)$$

και μετά τοποθετούνται με φορά αντίστροφη από αυτήν του ρολογιού. Έτσι το αποτέλεσμα μιας μάσκας 3χ3 είναι :

mask = [19 18 16 13 12 10 7
22 20 17 14 11 8 6
24 23 21 15 9 5 4

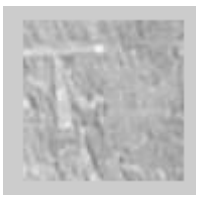
25 26 27 0 3 2 1
28 29 33 39 45 47 48
30 32 35 38 41 44 46
31 34 36 37 40 42 43] .

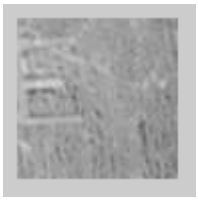
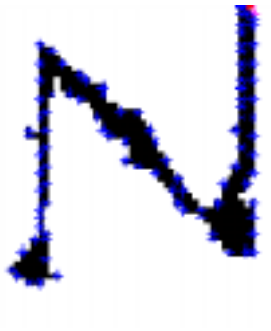
Οι αριθμοί δείχνουν απλά την φορά ψαξίματος η οποία προκύπτει απο τους αριθμούς που υπολογίζονται από τις παραπάνω εξισώσεις, αριθμώντας τους με αύξουσα σειρά. Το μεσαίο στοιχείο δεν παίζει κανέναν ρόλο στο ψάξιμο. Εμείς υποθέτουμε ότι M=N και ότι ο χαρακτήρας και το φόντο είναι μαύρα και άσπρα pixel αντίστοιχα.
Ο αλγόριθμος λειτουργεί ως εξής :

Βρίσκουμε το πρώτο μαύρο pixel του χαρακτήρα μας σκανάροντας την εικόνα μας από δεξιά προς τα αριστερά και από πάνω προς τα κάτω. Κεντράρουμε την μάσκα πάνω σε αυτόν τον χαρακτήρα και συνεχίζουμε το ψάξιμο σύμφωνα με την φορά της μάσκας. Μόλις βρούμε το επόμενο μαύρο pixel, το ψάξιμο συνεχίζεται αλλά από το pixel εκείνο που βρίσκεται 90^o δεξιά της τρέχουσας κατεύθυνσης (αναφορικά με το κέντρο). Αυτό βοηθάει το ψάξιμο να συνεχίζεται προς τα εμπρός και να αποφεύγονται επικαλύψεις περιοχών που έχουν ήδη ψαχτεί. Το ψάξιμο σταματάει μόλις βρεθεί το αρχικό pixel ή κάποιο pixel πολύ κοντινό σε αυτό.

3.1.1 Παραδείγματα εφαρμογής

Στον παρακάτω πίνακα φαίνονται μερικά παραδείγματα εφαρμογής του αλγορίθμου.

Αρχική Εικόνα	**Εξαγώμενος χαρακτήρας**	**Περίγραμμα**
		

3.1.2 Εύρεση ελάχιστης απόστασης μεταξύ των σπασμένων κοματιών με βάση τα σημεία του περιγράμματος

Χρειάστηκε να υλοποήσουμε τον παραπάνω αλγόριθμο γιατί επόμενο βήμα στην επεξεργασία των εικόνων είναι η εύρεση ελάχιστων αποστάσεων μεταξύ των κομματιών που έχουν απομείνει στην εικόνα μας. Υπολογίζοντας αυτές τις αποστάσεις χρησιμοποιώντας τα σημεία του περιγράμματος γλιτώνουμε χρόνο και απλοποιούμε αρκετά τα πράγματα. Στόχος μας είναι να «μαζέψουμε» τα κομμάτια των γραμμάτων που είναι σπασμένα και προφανώς θα βρίσκονται αρκετά κοντα, και να απομακρύνουμε περιοχες με απομονωμένα κομμάτια θορύβου.

Έτσι φτιάξαμε έναν αλγόριθμο που χρησιμοποιεί τα σημεία του περιγράμματος κάθε κομματιού της εικόνας και μας δίνει έναν πίνακα με τις ελάχιστες αποστάσεις των κομματιών μεταξύ τους.

Πιο συγκεκριμένα για κάθε σημείο του περιγράμματος ενός αντικειμένου υπολογίζουμε όλες τις αποστάσεις από τα σημεια του περιγράμματος των υπόλοιπων αντικειμένων και επιλέγουμε την μικρότερη κάθε φορά.

3.2 Αλγόριθμος □inimum Spanning Tree

Όπως αναφέρθηκε και προηγουμένως υλοποιήσαμε τον αλγόριθμο Minimum Spanning Tree για την συλλογή γειτονικών κομματιών από την εικόνα μας με την ελπίδα ότι αυτά θα ανήκουν σε σπασμένους χαρακτήρες. Ο αλγόριθμος αυτός δέχεται σαν είσοδο τον πίνακα που περιέχει τις ελάχιστες αποστάσεις μεταξύ των κομματιών και μας δίνει σαν έξοδο, λειτουργώντας κάτω από ένα thresh□ld απόστασης, κομμάτια που αποτελούν μια «γειτονιά». Πιο συγκεκριμένα, αν η απόσταση δυο αντικειμένων είναι μικρότερη από το Thresh□ld που έχουμε επιλέξει διαλέγουμε αυτά τα κομμάτια ως μέρη μιας «γειτονιάς». Αν η απόσταση των αντικειμένων ξεπερνάει την τιμή του Thresh□ld αποκόβουμε αυτά τα αντικείμενα από την «γειτονιά».

3.2.1 Παραδείγματα εφαρμογής

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα από τον συνδυασμό των αλγόριθμων που περιγράφηκαν παραπάνω.

Αρχική εικόνα	**Επεξεργασμένη Εικόνα**	**Περιοχές της εικόνας που επιλέγονται**
		 
		 

4.Αλγόριθμος Ge□metric □ □ments

4.1 Εισαγωγή στους τρόπους αναγνώρισης

Η αναγνώριση χαρακτήρων ανεξάρτητα από την θέση, το μέγεθος και την κατεύθυνσή τους αποτελεί κύριο στόχο τελευταίων ερευνών. Για να επιτευχθεί λειτουργικότητα και ευελιξία, οι μέθοδοι που χρησιμοποιούνται δεν θα πρέπει να επηρεάζονται από διακυμάνσεις των διαστάσεων των χαρακτήρων και θα πρέπει να παρέχουν βελτίωση της απόδωσης αναγνώρισης με επαναληπτικές δοκιμές.

Ένας από τους κεντρικούς στόχους σε αυτές τις μεθόδους είναι ο αυτόματος χαρακτηρισμός του περιεχομένου της εικόνας και η ανάκτηση των εικόνων βασισμένων στην ομοιότητα του περιεχομένου αυτού.

Για τον στόχο αυτό υπολογίζονται κάποια χαρακτηριστικά σχήματος **(shape measures)** η κατηγοριοποίηση των οποίων φαίνεται παρακάτω :

- **□utline based features:**
  - **(a)** ***Chain c□ded string*****:** □ κώδικας αλυσίδων, γνωστός και ως Freeman C□de, χρησιμοποιείται για να περιγράψει το περίγραμμα του αντικειμένου το οποίο αποτελεί και το μέτρο σύγκρισης για την αναγνώριση.
  - ***(b)*** ***F□urier descript□rs:*** Οι F□urier Descript□rs είναι μιγαδικοί συντελεστές που προκύπτουν από την επέκταση των κυματομορφών των σειρών F□urier.
  - ***(c)*** ***UNL F□urier descript□rs:*** Αυτά τα χαρακτηριστικά αποτελούν μια βελτιωμένη προέκταση των F□urier descript□rs.
- **Regi□n based features:**
  - **(a)** ***In□ariant m□ments***
  - **(b)** ***Zernike m□ments & pseud□-Zernike m□ments:*** Είναι χαρακτηριστικά που είναι ανεξάρτητα μόνο της περιστροφής και για αυτό χρησιμοποιούνται περισσότερο στην αναγνώριση σχημάτων.
- **C□mbined features:**

(a) ***□□ment in□ariant & F□urier descript□rs***

(b) ***□□ment in□ariant & UNL F□urier descript□rs***

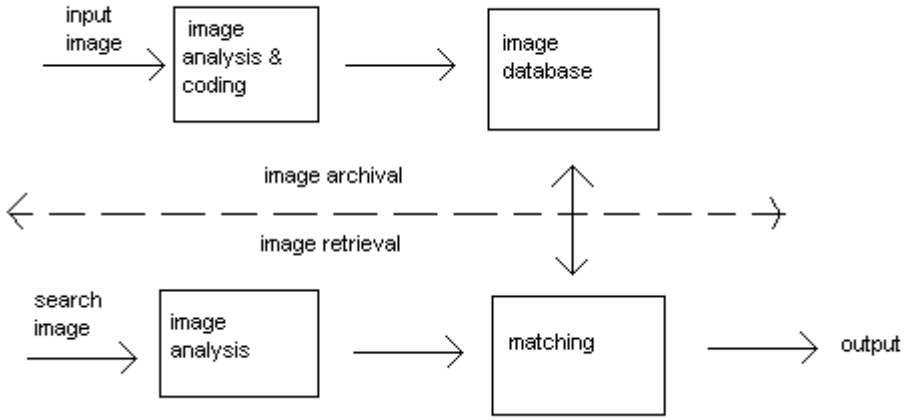
Αυτά τα χαρακτηριστικά είναι συνδυασμός των δύο μετρικών.

Από τις παραπάνω μεθόδους χρησιμοποιήσαμε την μέθοδο ***In□ariant □□ments***, γνωστή και ως μέθοδο των ροπών επειδή έχει το πλεονέκτημα να μπορεί να εφαρμοστεί και σε περιπτώσεις όπου η εικόνα περιέχει περισσότερα από ένα αντικείμενα, δηλαδή στις περιπτώσεις όπου ο χαρακτήρας είναι σπασμένος σε μικρότερα κομμάτια. Αυτήν την δυνατότητα δεν την παρέχουν οι υπόλοιπες μέθοδοι αφού εφαρμόζονται σε εικόνες όπου περιέχουν ένα αντικείμενο.

4.2 Ανάκτηση εικόνων

Το πρόβλημά μας έχει ως εξής : Έχοντας σαν είσοδο μια εικόνα που περιέχει χαρακτήρα προς αναγνώριση θα θέλαμε να βρούμε σε ποια από τις εικόνες που έχουμε ως πρότυπα μοιάζει περισσότερο. Για να το λύσουμε αυτό χρειαζόμαστε δύο πράγματα. Πρώτα κάποια χαρακτηριστικά που θα αποτελούν την πληροφορία που αντιπροσωπεύει την εικόνα και θα χρησιμοποιηθούν ως μέτρο σύγκρισης και δεύτερον ενα κριτήριο σύγκρισης της ομοιότητας . Τα χαρακτηριστικά λοιπόν που χρησιμοποιούμε είναι οι κεντρικές ροπές και το κριτήριο είναι η απόσταση ομοιότητας . Και τα δύο αυτά θέματα παρουσιάζονται παρακάτω.

Στο επόμενο σχήμα παρουσιάζεται μια γενική εικόνα του μοντέλου ανάκτησης και αναγνώρισης των χαρακτήρων.



Οι εικόνες εισόδου, οι οποίες είναι εικόνες με τα γράμματα της αλφαβήτου που χρησιμοποιούνται σαν πρότυπα, αναλύονται για να εξάγουμε τα χαρακτηριστικά τους τα οποία και αποθηκεύουμε μαζί με αυτές. Όποτε εισάγεται μια εικόνα για ψάξιμο, αναλύεται και εξάγονται από αυτήν και τα δικά της χαρακτηριστικά τα οποία συγκρίνονται με αυτά των προτύπων . Εδώ πρέπει να αναφέρουμε ότι λόγω της ιδιαιτερότητας των χαρακτήρων μας, χρειάστηκε για κάθε γράμμα να κρατήσουμε περισσότερα από ένα πρότυπα.

4.3 Υλοποίηση αλγορίθμου

4.3.1 Η μέθοδος των ροπών

Οι ροπές έχουν χρησιμοποιηθεί σε έναν αριθμό εφαρμογών προκειμένου να χρησιμοποιηθούν ως αμετάβλητα χαρακτηριστικά σε προβλήματα αναγνώρισης διδιάστατων προτύπων. Οι κανονικές ροπές m_{pq} ενός διδιάστατου προτύπου (εικόνας) το οποίο συμβολίζεται με την συνάρτηση f (x,y), μπορεί να ορισθούν ως ακολούθως :

$$\mu_{pq}=\Sigma_x\Sigma_y\, x^p * y^q * f(x, y), \qquad p, q= 0, 1, 2 \ldots$$

 Hu πρώτος εισήγαγε τις ροπές σαν χαρακτηριστικά αναγνώρισης εικόνων. Χρησιμοποιώντας μη-γραμμικούς συνδιασμούς των κανονικοποιημένων κεντρικών ροπών , παρήγαγε ένα σύνολο από επτά αμετάβλητες ροπές **(invariant m ments)** οι οποίες έχουν τις επιθυμητές ιδιότητες να είναι αμετάβλητες ως προς την περιστροφή. Πιο

ειδικά , οι κεντρικές ροπές έχουν την ιδιότητα ότι είναι ανεξάρτητες της μετατόπισης και δίνονται από τους τύπους :

$$\mu_{pq} = \Sigma_x\Sigma_y(x- \bar{x})^p *(y- \bar{y})^q , \qquad p, q= 0, 1, 2 \ldots$$

όπου

$\bar{x} = m_{10}/m_{00}$ **και** $\bar{y} = m_{01}/m_{00}$ είναι ουσιαστικά το συνολικό κέντρο βάρους των αντικειμένων της εικόνας.

Οι επόμενες ροπές $\varphi_1,\varphi_2,...,\varphi_7$ είναι ανεξάρτητες της θέσης και της περιστροφής:

$$\varphi_1= \mu_{20} + \mu_{02}$$

$$\varphi_2= (\mu_{20}-\mu_{02})^2 +4\mu_{11}^2$$

$$\varphi_3= (\mu_{30}-3\mu_{12})^2 + (3\mu_{21}-\mu_{03})^2$$

$$\varphi_4= (\mu_{30}+\mu_{12})^2 + (3\mu_{21}+\mu_{03})^2$$

$$\varphi_5=(\mu_{30}-3\mu_{12})* (\mu_{30}+\mu_{12})*[(\mu_{30}+\mu_{12})^2 – 3*(\mu_{21}+\mu_{03})^2] +$$
$$+ (3\mu_{21}-\mu_{03})*(\mu_{21}+\mu_{03})*[3*(\mu_{30}+\mu_{12})^2 – (\mu_{21}+\mu_{03})^2]$$

$$\varphi_6=(\mu_{20}-\mu_{02})*[(\mu_{30}+\mu_{12})^2 - (\mu_{21}+\mu_{03})^2] +4*\mu_{11}*(\mu_{30}+\mu_{12})*(\mu_{21}+\mu_{03})$$

$$\varphi_7=(3\mu_{21}-\mu_{03}) *(\mu_{30}+\mu_{12})*[(\mu_{30}+\mu_{12})^2 – 3*(\mu_{21}+\mu_{03})^2] –$$
$$- (\mu_{30}-3\mu_{12}) *(\mu_{21}+\mu_{03})*[3*(\mu_{30}+\mu_{12})^2 - (\mu_{21}+\mu_{03})^2]$$

Οι παραπάνω ροπές μπορεί να κανονικοποιηθούν έτσι ώστε να γίνουν ανεξάρτητες της κλιμάκωσης , αντικαθιστώντας τις κεντρικές ροπές μ_{pq} με τις κανονικοποιημένες κεντρικές ροπές η_{pq}, στις παραπάνω εξισώσεις. Οι κανονικοποιημένες κεντρικές ροπές ορίζονται ως ακολούθως :

$$\eta_{pq}= \mu_{pq}/ \mu_{00}^{\gamma} \quad \text{όπου } \gamma= (p+q) / 2 +1 \quad \text{για } p+q = 2,3,...$$

Αφού έχουμε υπολογίσει τις ροπές γαι την εικόνα που μας ενδιαφέρει το σημαντικό είναι να καθορίσουμε την ομοιότητά της με κάποια από τις εικόνες που έχουμε ως πρότυπα.

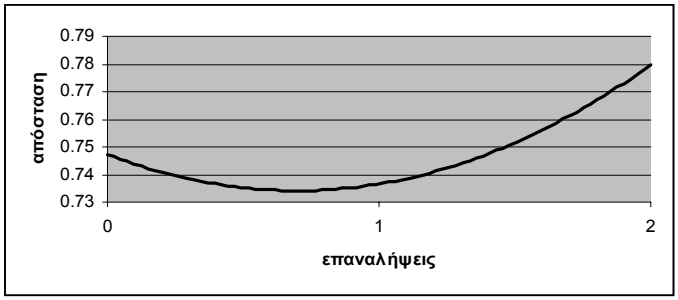
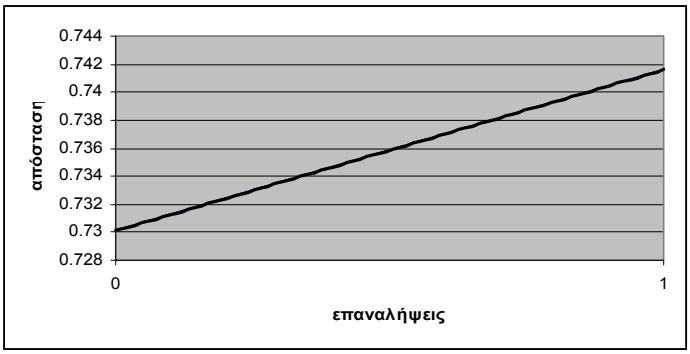
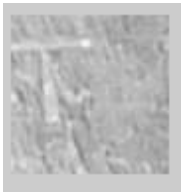
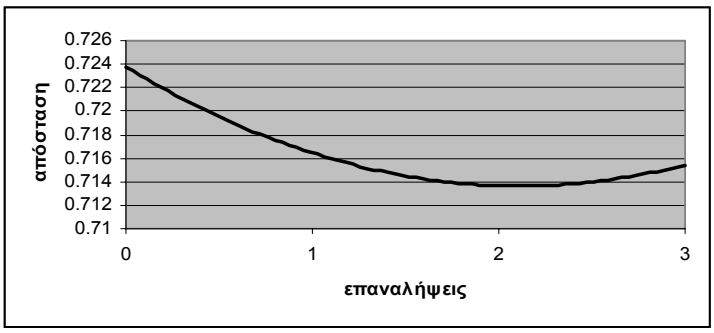
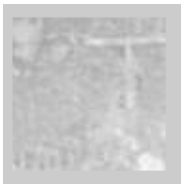
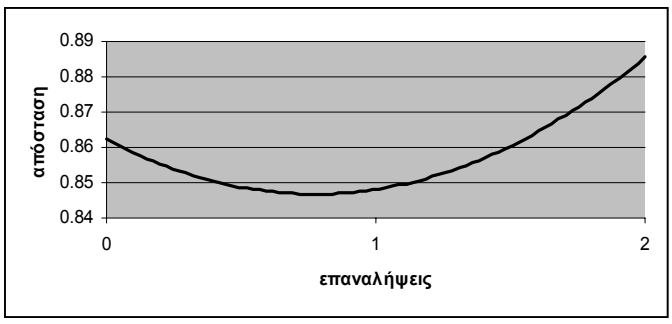
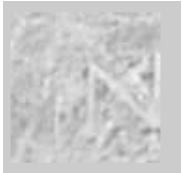
Η μετρική σύγκρισης που χρησιμοποιήσαμε είναι η Ευκλείδια απόσταση όπου για δύο πίνακες χαρακτηριστηκών M^Q και M^I για ένα ζευγάρι εικόνων Q και I υπολογίζεται ως εξής:

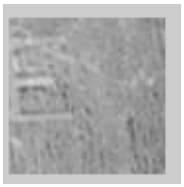
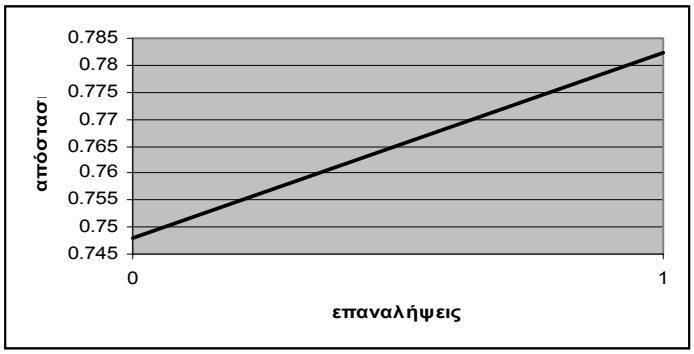
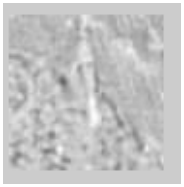
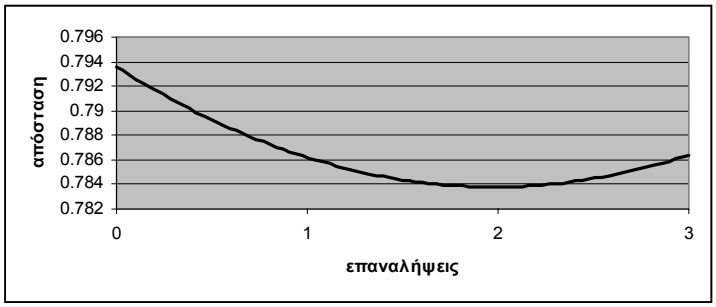
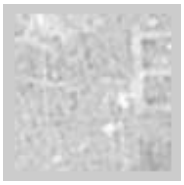
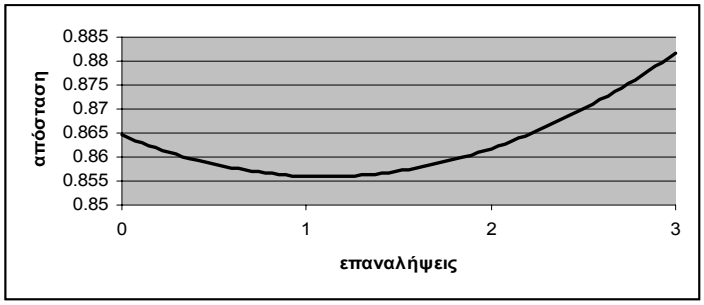
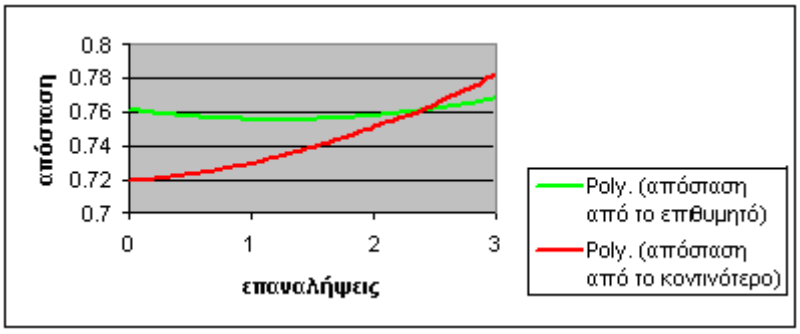
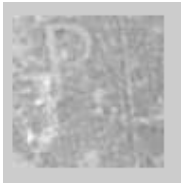
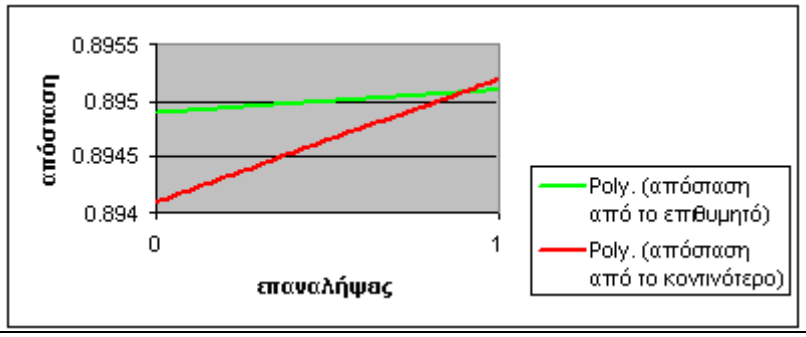
$$\mathbf{d_m (M^Q , \square^I) = sqrt (\Sigma_{i=1}^{i=7} (m_i^Q - m_i^I)^2)}$$

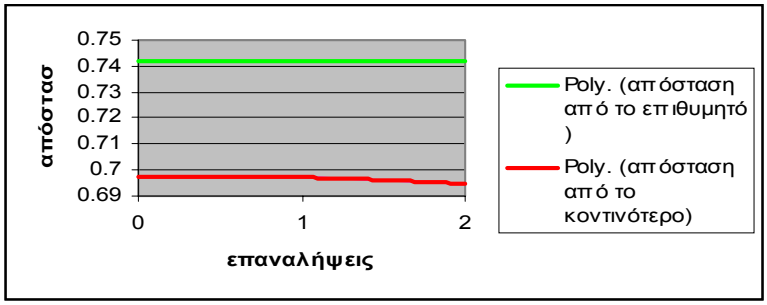
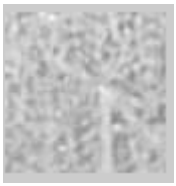
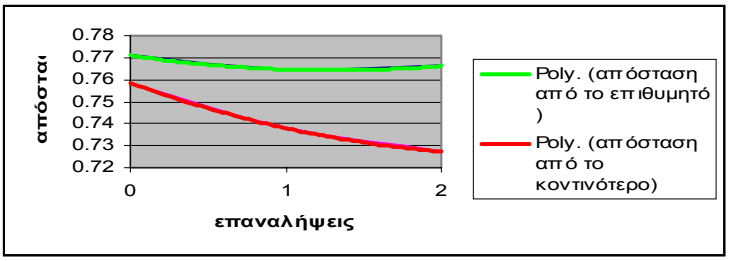
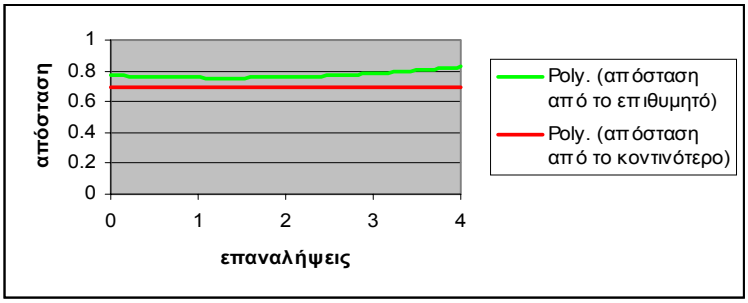
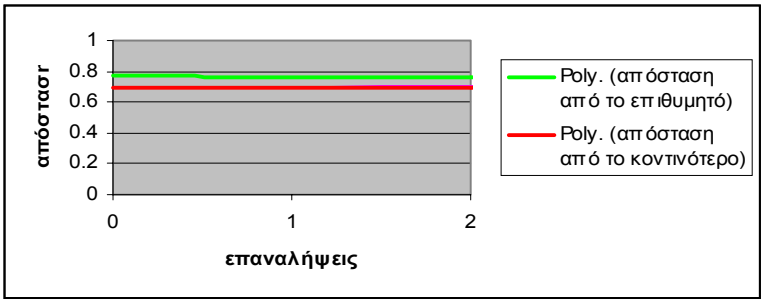
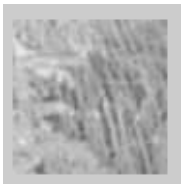
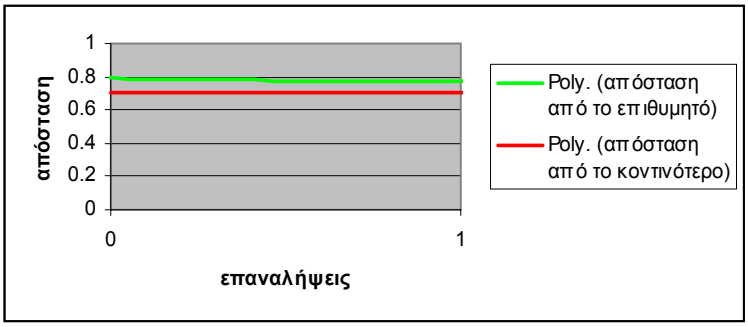
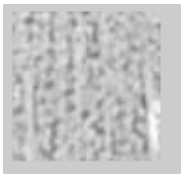
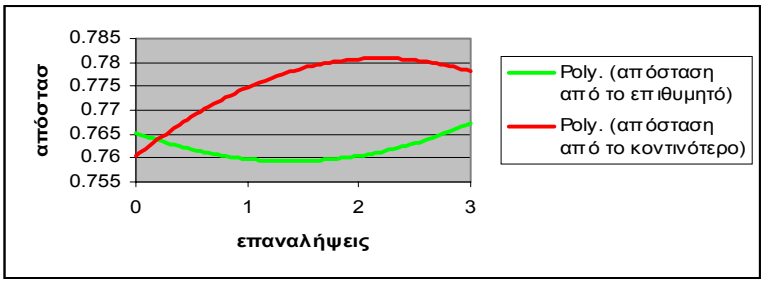
Η τιμή της απόστασης ομοιότητας $\mathbf{d_m}$ είναι μηδέν ή πολύ μικρή για παρόμοιους χαρακτήρες . Έτσι κρατήσαμε τις αποστάσεις της εικόνας από κάθε χαρακτήρα πρότυπο με αύξουσα σειρά έτσι ώστε η πρώτη μικρότερη τιμή να αντιπροσωπεύει τον χαρακτήρα στον οποίο μοιάζει η εικόνα μας .

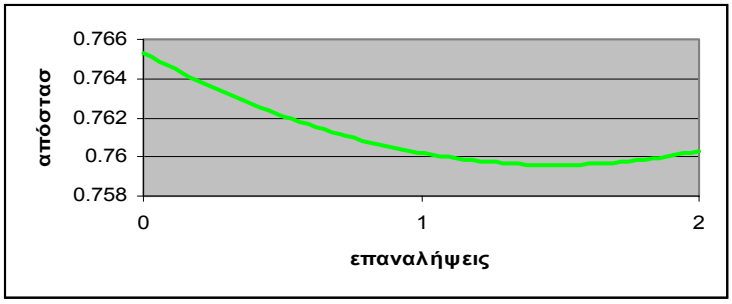
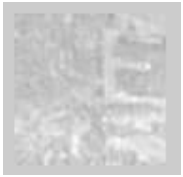
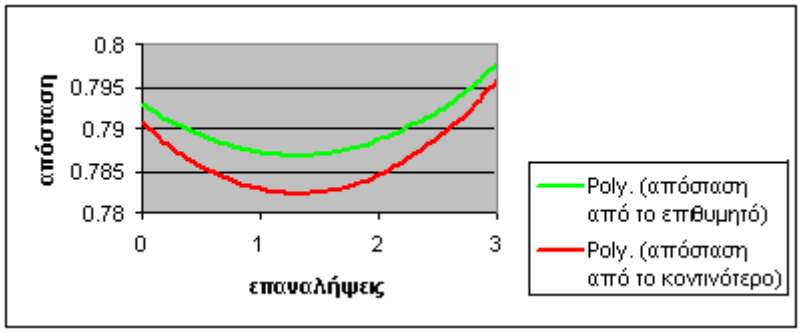
4.3.2 Αποτελέσματα εφαρμογής αλγορίθμου

Εφαρμόσαμε τον αλγόριθμο στις εικόνες που προέκυψαν από την επεξεργασία με τις μεθόδους που προαναφέρθησαν στα άλλα κεφάλαια και τα αποτελέσματα φαίνονται στον παρακάτω πίνακα. Σημαντικό ρόλο στα αποτελέσματα αυτά έπαιξε η τιμή του thresh□ld για την μετατροπή της εικόνας σε ασπρόμαυρη. Έτσι μετά τη πρώτη εφαρμογή αυξάναμε το thresh□ld κατά ένα πολύ μικρό ποσοστό δ, αφού μικρές διαφορές στην τιμή του επηρεάζουν σημαντικά την εικόνα , και ξαναεφαρμόσαμε τον αλγόριθμο μέχρι να πετύχουμε καλύτερα αποτελέσματα που αυτό στην περίπτωσή μας σημαίνει μείωση της απόστασης ομοιότητας από τον χαρακτήρα.

Αρχική εικόνα	Κοντινότερο γράμμα	Μεταβολή των αποστάσεων ομοιότητας από το κοντινότερο και από το επιθυμητό γράμμα
1. 	O 	
2. 	E 	
3. 	T 	
4. 	T 	
5. 	N 	

6. 	**E** ✓	
7. 	**I** ✓	
8. 	**E** ✓	
9. 	**Z** ✗	
10. 	**B** ✗	

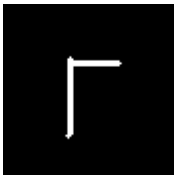
11. 	**Θ** 	
12. 	**Γ** 	
13. 	**Θ** 	
14. 	**Η** 	
15. 	**Η** 	
16. 	**Γ** 	

17. 	**O** 	
18. 	**I** 	

4.3.3 Σχολιασμός των αποτελεσμάτων

Στις περιπτώσεις όπου το classificati□n γινόταν σωστά τις περισσότερες φορές η μείωση του Thresh□ld μείωνε και την απόσταση ομοιότητας μέχρι ενός σημείου διατηρώντας και την σωστή αντιστοίχιση.

Στις περιπτώσεις όπου το classificati□n μας έδινε λάθος αποτέλεσμα είχαμε διάφορα «σενάρια» όπως στην **περίπτωση 16** όπου ενώ αρχικά ο χαρακτήρας μας που είναι το Ι έμοιαζε περισσότερο με το Γ από τα πρότυπά μας :

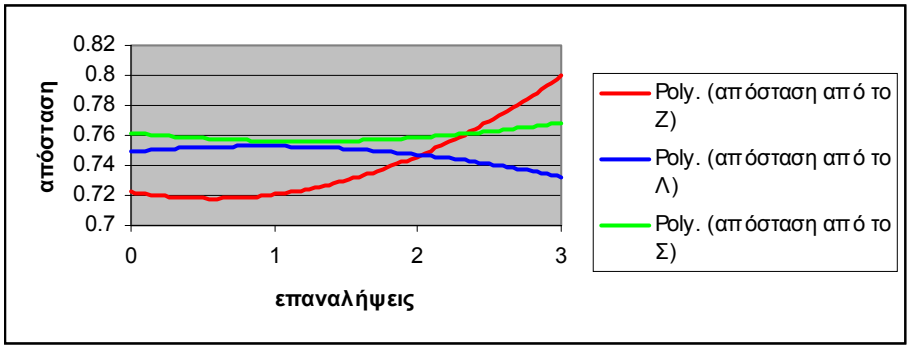
Ενώ από την πρώτη αύξηση και μετά έμοιαζε στο Ι:

Μια διαφορετική περίπτωση είναι η **περίπτωση 9** όπου αρχικά και μετά τις δύο πρώτες αυξήσεις το Σ της εικόνας μας έμοιαζε με το Z από τα πρότυπα:


Ενώ στην τρίτη αύξηση ο χαρακτήρας μας έμοιαζε με το Λ:

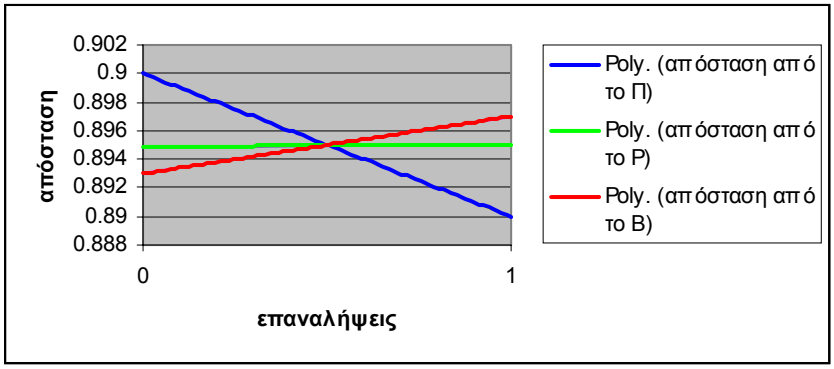

Έτσι παρακάτω φαίνονται οι διακυμάνσεις και των τριών αποστάσεων:



Παρόμοια είναι και η **περίπτωση 10** όπου αρχικά ο χαρακτήρας μας Ρ μοιάζει με το Β περισσότερο:


Ενώ με την πρώτη αύξηση μοιάζει με το Π:

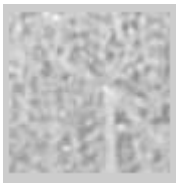
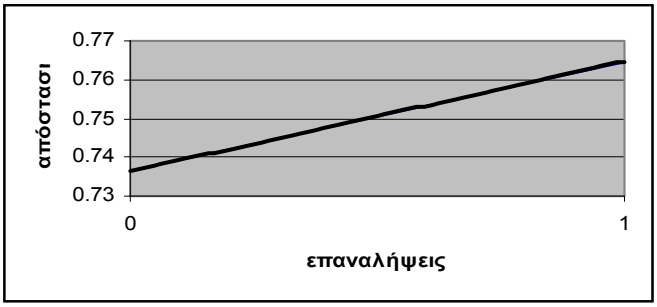
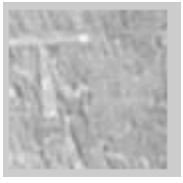
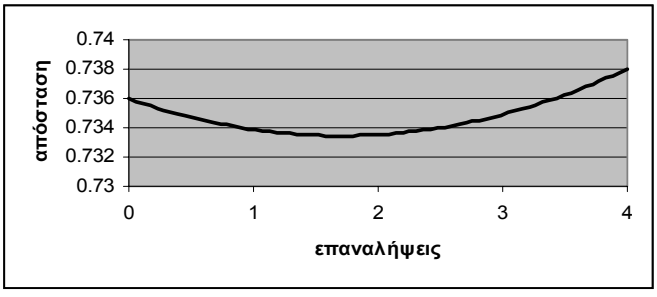

Έτσι παρακάτω φαίνονται οι διακυμάνσεις και των τριών αποστάσεων:

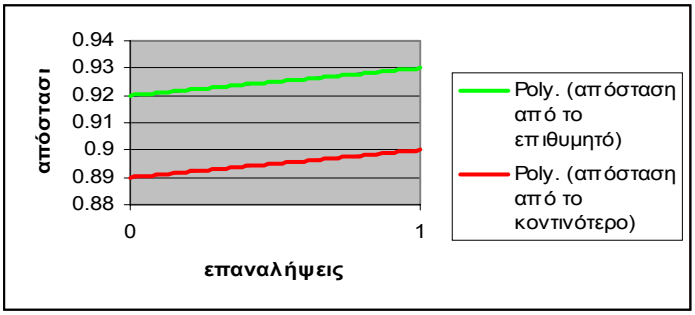
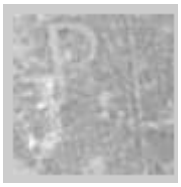
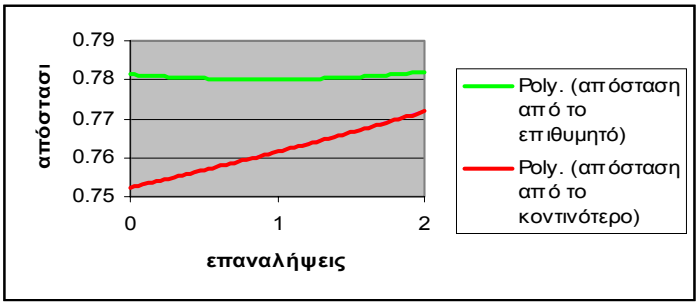
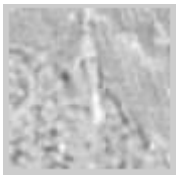
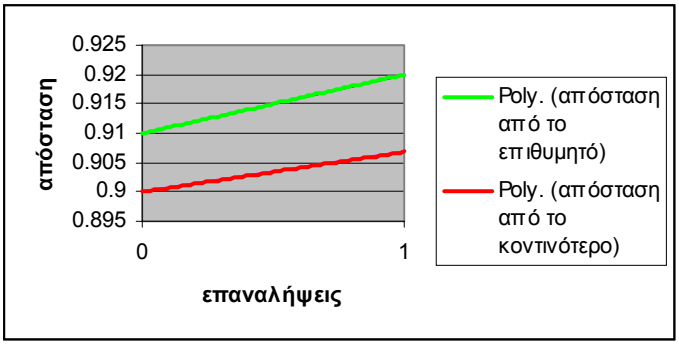
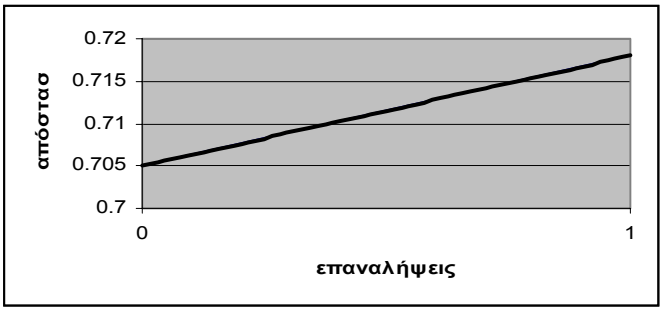
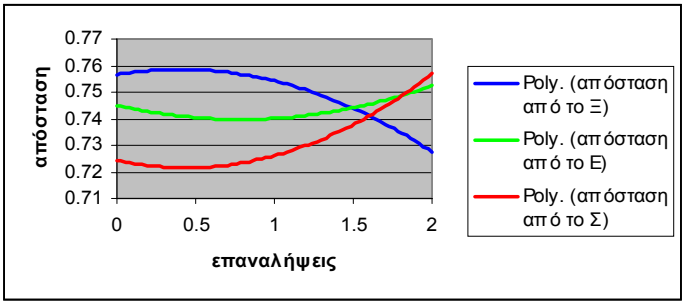


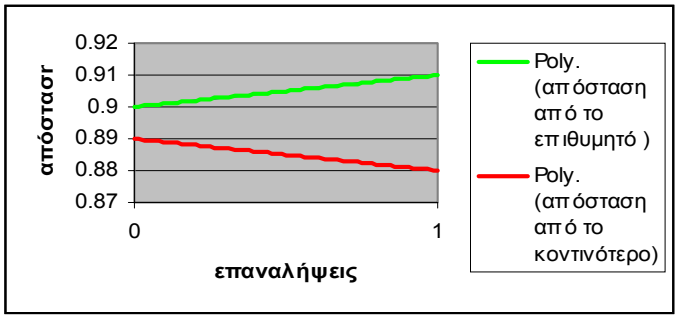
Στις υπόλοιπες περιπτώσεις όπου το Classificati□n έγινε λάθος παρά την μείωση της απόστασης από τον επιθυμητό χαρακτήρα το κοντινότερο μοντέλο συνέχισε να είναι το αρχικό λανθασμένο μοντέλο.

4.3.4 Η μέθοδος των ροπών στις ασπρόμαυρες εικόνες που προκύπτουν από το Fit

Εφαρμόσαμε τον αλγόριθμο ge□metric m□ments και στις εικόνες που προκύπτουν από το φιτ και τις έχουμε επεξεργαστεί με τεχνικές m□rph□l□gy. Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα:

Αρχική εικόνα	**Κοντινότερο γράμμα**	**Μεταβολή των αποστάσεων ομοιότητας από το κοντινότερο και από το επιθυμητό γράμμα**
1. 	**A** √	
2. 	**T** √	

3. 	**Γ** ✗	 απόστασι 0.94 0.93 0.92 0.91 0.9 0.89 0.88 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
4. 	**Ε** ✗	 απόστασι 0.79 0.78 0.77 0.76 0.75 0 1 2 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
5. 	**Γ** ✗	 απόσταση 0.925 0.92 0.915 0.91 0.905 0.9 0.895 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
6. 	**Ν** ✓	 απόσταο 0.72 0.715 0.71 0.705 0.7 0 1 επαναλήψεις
7. 	**Σ** ✗	 απόσταση 0.77 0.76 0.75 0.74 0.73 0.72 0.71 0 0.5 1 1.5 2 επαναλήψεις Poly. (απόσταση από το Ξ) Poly. (απόσταση από το Ε) Poly. (απόσταση από το Σ)

8. 	**Γ** 	

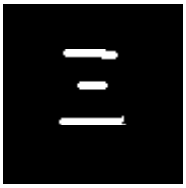
4.3.5 Σχολιασμός των αποτελεσμάτων

Από τα παραπάνω βλέπουμε ότι όπως και με τις προηγούμενες εικόνες έτσι και εδώ έχουμε περιπτώσεις όπου το classificati□n γίνεται σωστά και με την αύξηση του Thresh□ld η απόσταση ομοιότητας από το επιθυμητό γράμμα μικραίνει όπως η 2, περιπτώσεις όπου ενώ έχουμε σωστό classificati□n η αύξηση του Thresh□ld αυξάνει και την διαφορά ομοιότητας όπως η 1 και η 6, περιπτώσεις όπου το classificati□n γίνεται λάθος και κοντινότερη απόσταση παραμένει αυτή του λανθασμένου προτύπου παρά την αύξηση του Thresh□ld όπως οι 3,4,5 και 8, και περιπτώσεις όπου το κοντινότερο μοντέλο αλλάζει με την αύξηση του Thresh□ld όπως η **περίπτωση 7** που αναλύεται παρακάτω μιας και παρουσιάζει το μεγαλύτερο ενδιαφέρον.

Ενώ αρχικά και μετά την πρώτη αύξηση του Thresh□ld το κοντινότερο μοντέλο ήταν το Σ:

μετά την $2^{η}$ αύξηση το κοντινότερο μοντέλο ήταν το Ξ, πράγμα που είναι λογικό όπως φαίνεται παρακάτω :

Σε αυτες τις περιπτώσεις ίσως είχαμε μεγαλύτερο ποσοστό αποτυχίας αφού η βάση μας ήταν περισσότερο προσαρμοσμένη στους χαρακτήρες που εξάγαμε με τις προηγούμενες μεθόδους .

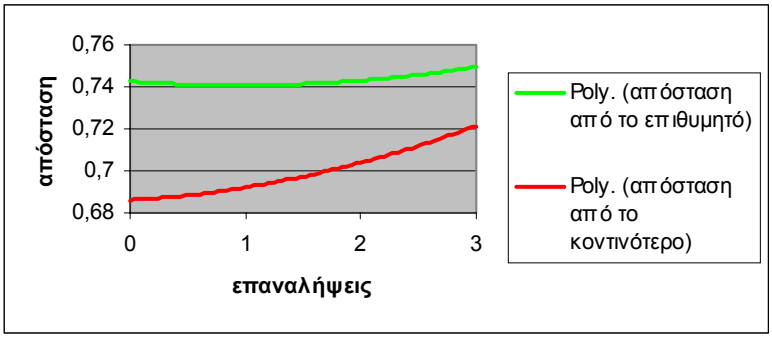
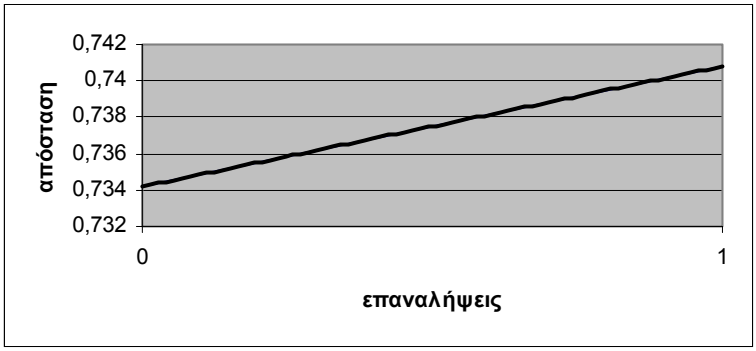
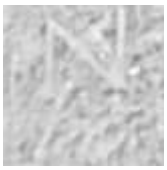
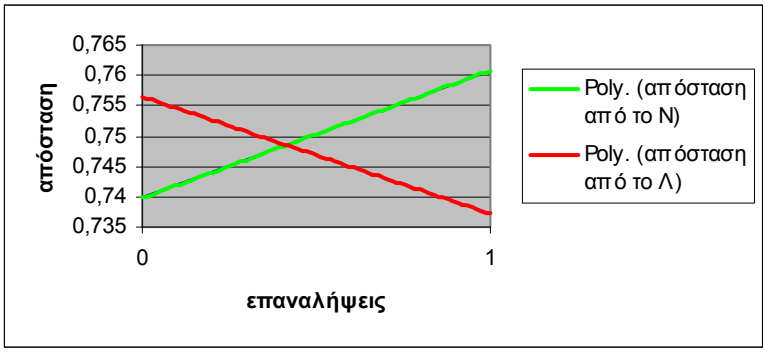
4.4 φαρμογή του αλγορίθμου με διαφορετικο κόψιμο της εικόνας

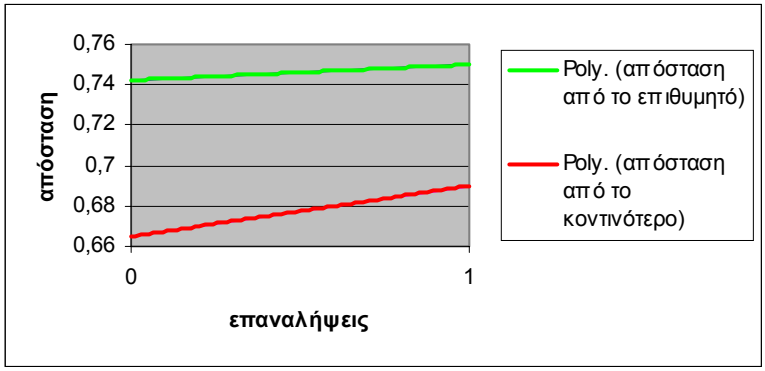
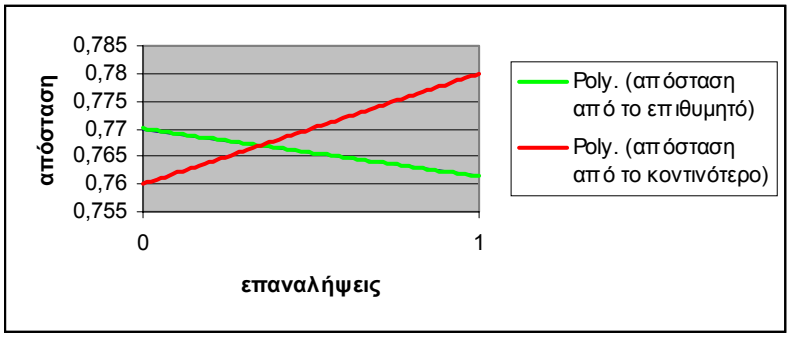
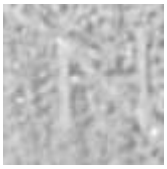
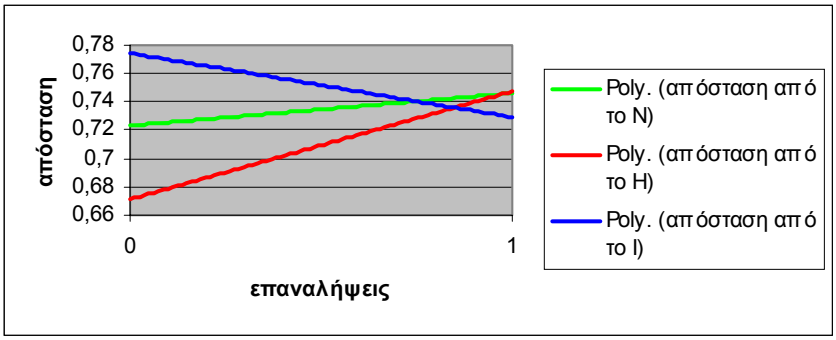
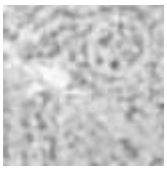
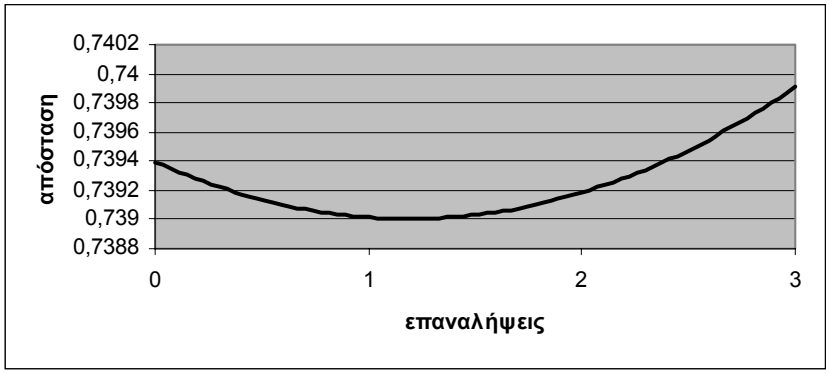
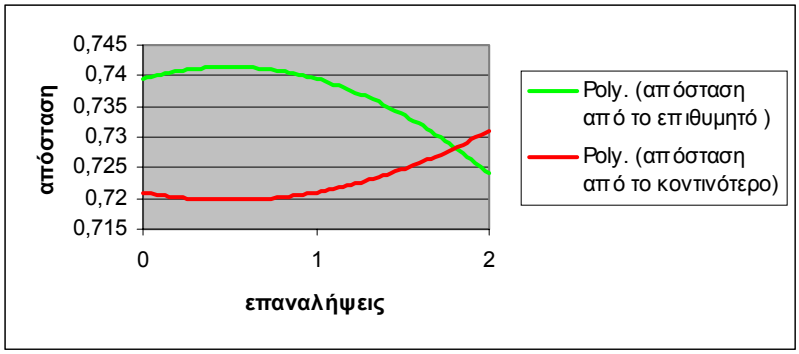
Εφαρμόσαμε τον αλγόριθμο στην αρχική μας εικόνα αφού πρώτα την κάναμε circular shift έτσι ώστε να η διαίρεσή της τώρα στα 48 καινούρια κομμάτια να είναι διαφορετική. ¨Ετσι η εικόνα μας :

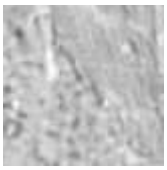
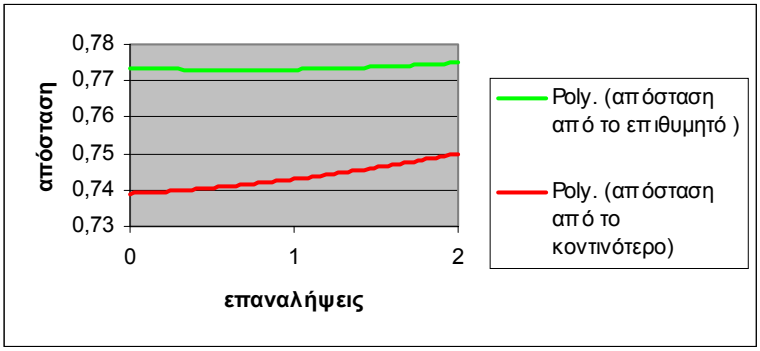
Έγινε:

Όπως φαίνεται παραπάνω ξαναέγινε η διαίρεση της Shifted εικόνας σε 48 καινούρια κομμάτια. Σε πολλές περιπτώσεις τα γράμματα που είχαμε ολόκληρα με το προηγούμενο κόψιμο κόπηκαν ενώ σε άλλες εμφανίστηκαν μέρη γραμμάτων που με το προηγούμενο κόψιμο είχαν χαθεί.

Οι καινούριες εικόνες περάσαν πάλι από την επεξεργασία με την μέθοδο του Fit εφαρμοστεί σε αυτές ο αλγόριθμος Ge□metric M□ments και τα τελικά αποτελέσματα του αλγορίθμου φαίνονται στον παρακάτω πίνακα:

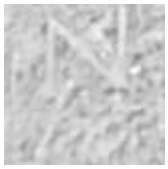
Αρχική εικόνα	**Κοντινότερο γράμμα**	**Μεταβολή των αποστάσεων ομοιότητας από το κοντινότερο και από το επιθυμητό γράμμα**
1. 	**Θ** ✗	
2. 	**Ο** √	
3. 	**Ν** √	

4. 	**Θ** ✗	 απόσταση 0,76 0,74 0,72 0,7 0,68 0,66 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
5. 	**T** ✗	 απόσταση 0,785 0,78 0,775 0,77 0,765 0,76 0,755 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
6. 	**H** ✗	 απόσταση 0,78 0,76 0,74 0,72 0,7 0,68 0,66 0 1 επαναλήψεις Poly. (απόσταση από το N) Poly. (απόσταση από το H) Poly. (απόσταση από το I)
7. 	**O** ✓	 απόσταση 0,7402 0,74 0,7398 0,7396 0,7394 0,7392 0,739 0,7388 0 1 2 3 επαναλήψεις
8. 	**Ψ** ✗	 απόσταση 0,745 0,74 0,735 0,73 0,725 0,72 0,715 0 1 2 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)

9. 	**Y** 	

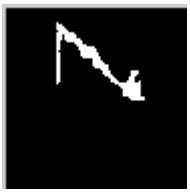
4.4.1 Σχολιασμός των αποτελεσμάτων

Περίπτωση 3:



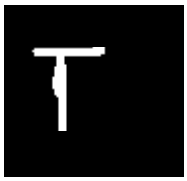
Κατά την πρώτη εφαρμογή του αλγορίθμου το classificati□n έγινε σωστά αντιστοιχίζοντας στον χαρακτήρα μας το N:

Μετά την πρώτη αύξηση όμως του thresh□ld ο αλγόριθμος μας έβγαλε ως κοντινότερο γράμμα το Λ, πράγμα που είναι λογικό όπως φαίνεται παρακάτω:



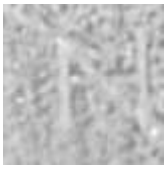
Περίπτωση 5:

Σε αυτήν την περίπτωση ενώ αρχικά το classificati□n μας έδινε ως κοντινότερο χαρακτήρα το T :

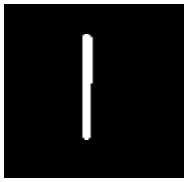
Μετά την πρώτη αύξηση κοντινότερος χαρακτήτρας ήταν το Ι όπως θε έπρεπε:


Περίπτωση 6:



Εδώ ο χαρακτήρας μας που , που είναι το Ν, με την πρώτη εφαρμογή του αλγορίθμου έμοιαζε στο Η:

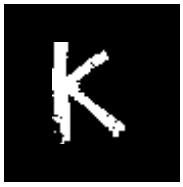

Ενώ μετά την πρώτη αύξηση του thresh□ld το κοντινότερο μοντέλο ήταν το Ι:

Περίπτωση 8:

Για τις δύο πρώτες επαναλήψεις του αλγορίθμου κοντινότερο μοντέλο στον χαρακτήρα μας ήταν το Ψ:

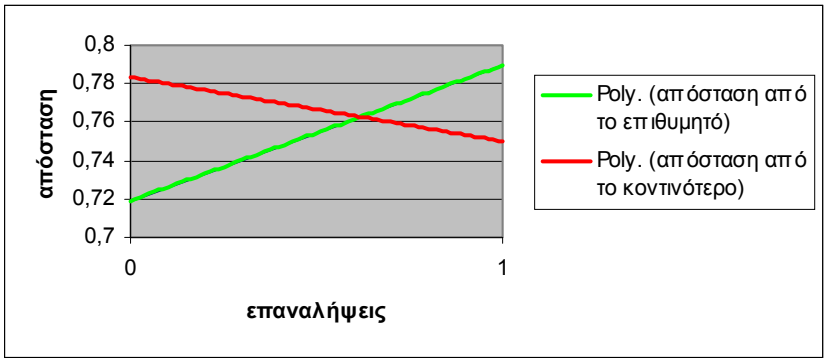
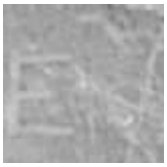
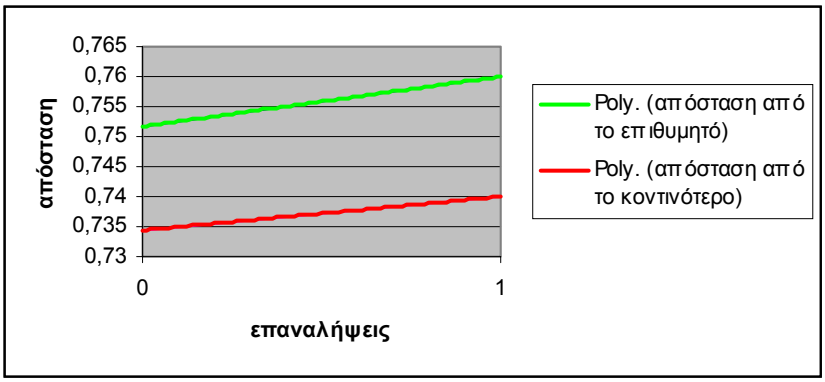
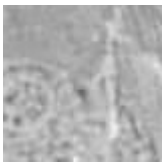
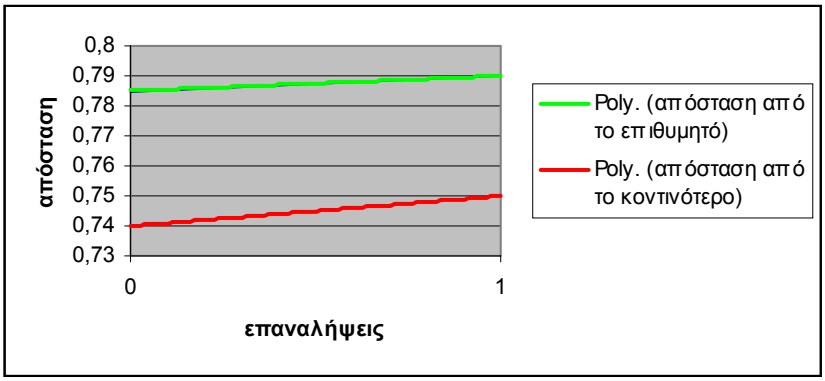
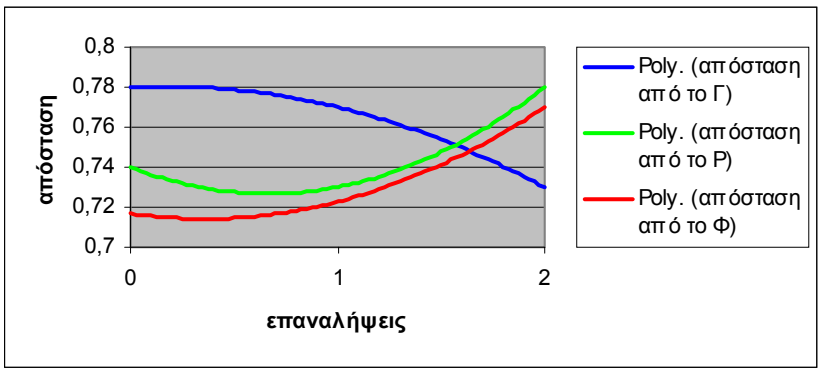

Ενώ με την Τρίτη αύξηση του Thresh□ld ο κοντινότερος χαρακτήρας ήταν το Κ:

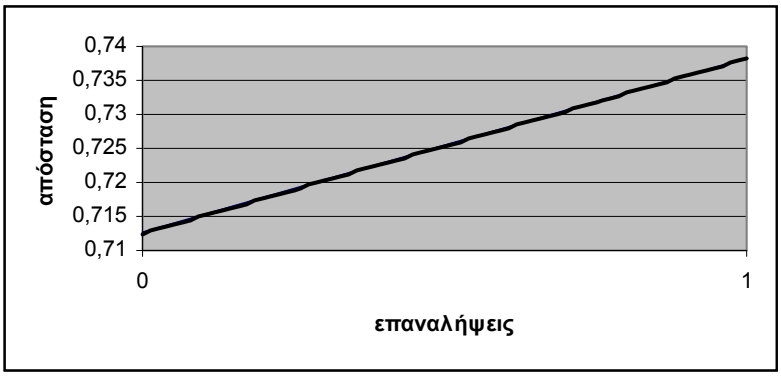
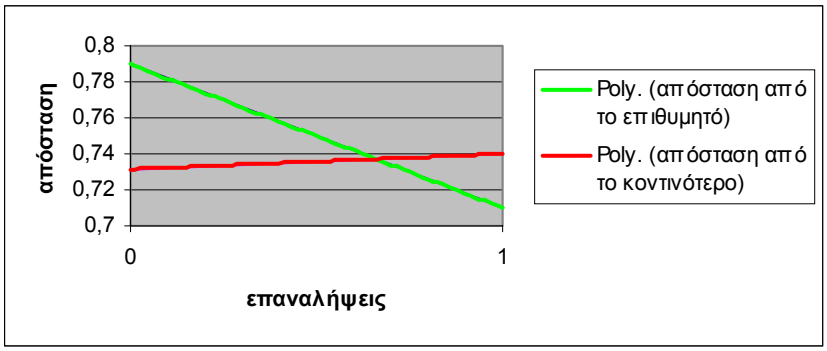
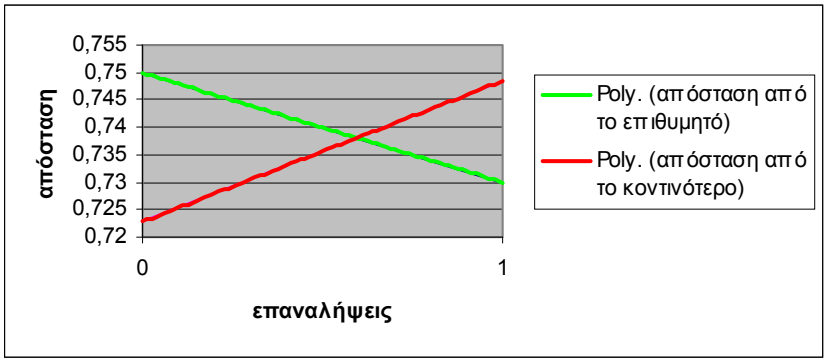
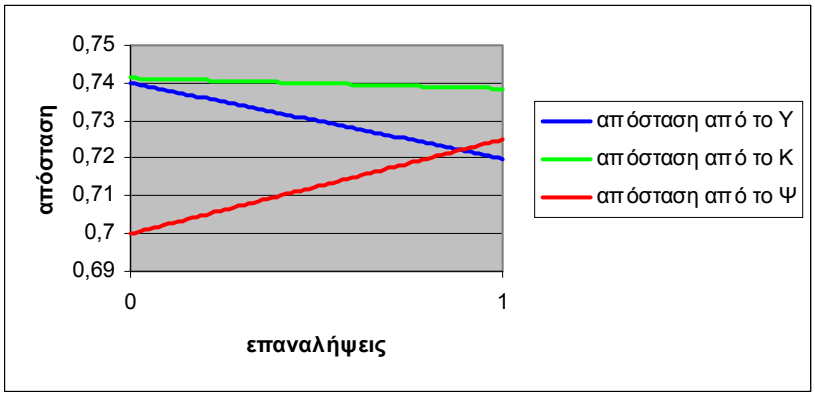



4.5 □φαρμογή του αλγορίθμου στην δεύτερη εικόνα

Η δεύτερη εικόνα που είχαμε στην διάθεση μας και φαίνεται παρακάτω, πέρασε από τις ίδιες επεξεργασίες με την πρώτη και έτσι στον πίνακα που ακολουθεί φαίνονται μερικά χαρακτηριστικά παραδείγματα:

Εικόνα 2:Η δεύτερη εικόνα της μαρμάρινης επιγραφής στην οποία εφαρμόστηκε ο αλγόριθμος.. Στην εικόνα αυτή φαίνεται και πως ακριβώς έχει γίνει το segmentati□n.

Αρχική εικόνα	Κοντινότερο γράμμα	Μεταβολή των αποστάσεων ομοιότητας από το κοντινότερο και από το επιθυμητό γράμμα
1. 	Τ ✓	
2. 	Β ✗	
3. 	Τ ✗	
4. 	Φ ✗	

5. 		 απόσταση 0,74 0,735 0,73 0,725 0,72 0,715 0,71 0 1 επαναλήψεις
6. 		 απόσταση 0,8 0,78 0,76 0,74 0,72 0,7 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
7. 	**B** 	 απόσταση 0,755 0,75 0,745 0,74 0,735 0,73 0,725 0,72 0 1 επαναλήψεις Poly. (απόσταση από το επιθυμητό) Poly. (απόσταση από το κοντινότερο)
8. 	**Ψ** 	 απόσταση 0,75 0,74 0,73 0,72 0,71 0,7 0,69 0 1 επαναλήψεις απόσταση από το Υ απόσταση από το Κ απόσταση από το Ψ

4.5.1 Σχολιασμός των αποτελεσμάτων

Περίπτωση 1:

Με την πρώτη εφαρμογή του αλγορίθμου ο κοντινότερος χαρακτήρας ήταν το Τ:

Ενώ με την πρώτη αύξηση του Thresh□ld το classificati□n αλλάζει και μας δίνει κοντινότερο χαρακτήρα το Γ:

Περίπτωση 4:

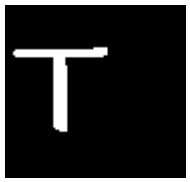
Αρχικά το Ρ της εικόνας μας έμοιαζε περισσότερο με το Φ:

Ενώ με την πρώτη αύξηση του Thresh□ld κοντινότερος χαρακτήρας ήταν το Γ:

Περίπτωση 6:

Το πρώτο classificati□n μας έδωσε κοντινότερο χαρακτήρα το X:

Ενώ με την αύξηση του Thresh□ld το κομμάτι θορύβου αποκολήθηκε και έγινε σωστά η αντιστοίχηση με το T:



Περίπτωση 7:

Όπως και στην προηγούμενη περίπτωση ενώ αρχικά το classificati□n έγινε λάθος, δίνοντας μας κοντινότερο γράμμα το B

Μετά την πρώτη αύξηση του thresh□ld η αντιστοίχιση έγινε σωστά με το E:

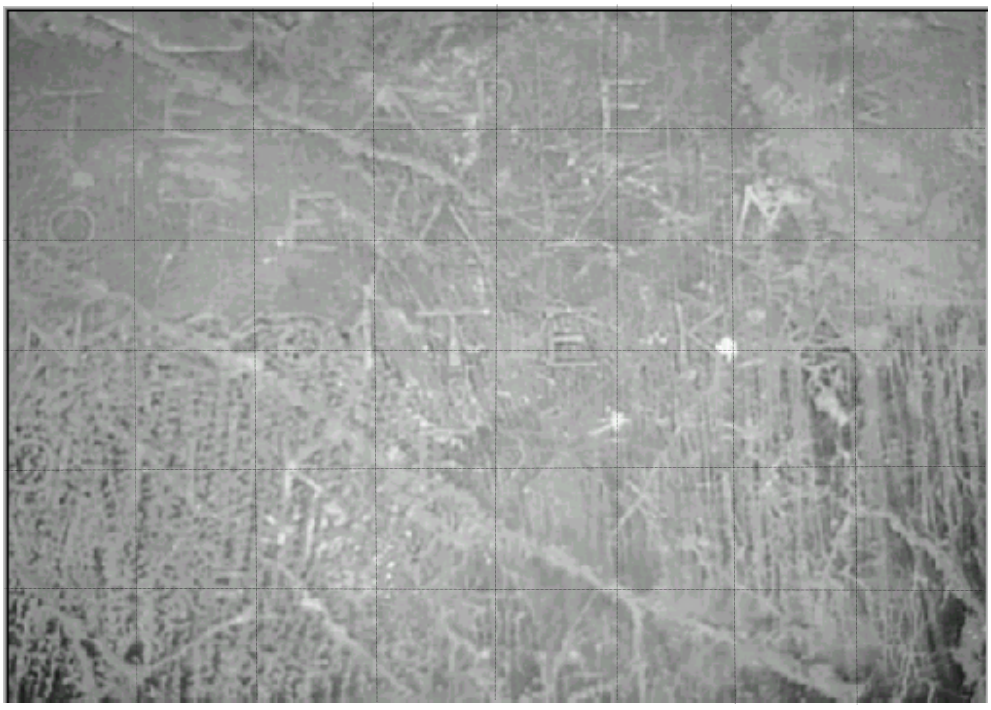
Περίπτωση 8:

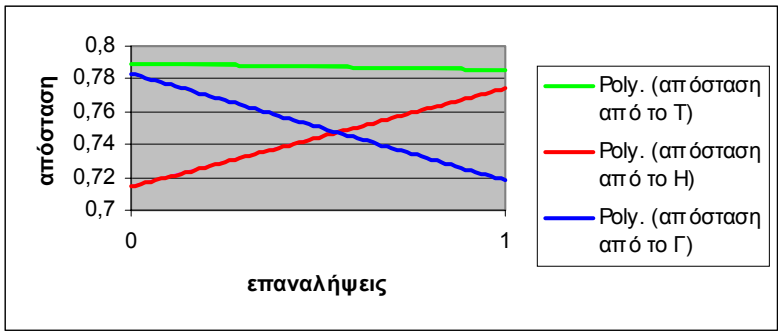
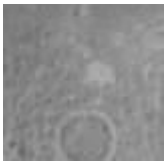
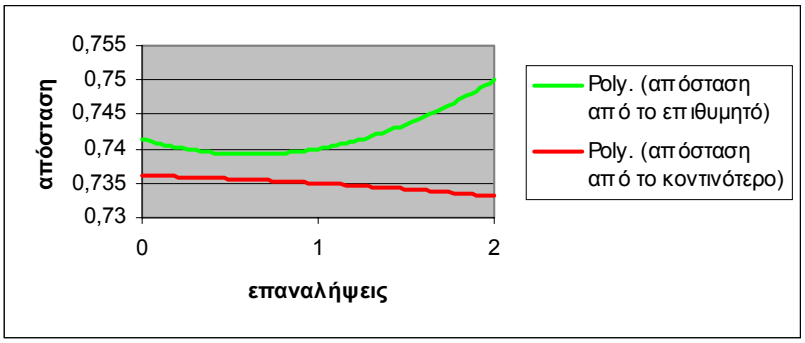
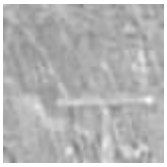
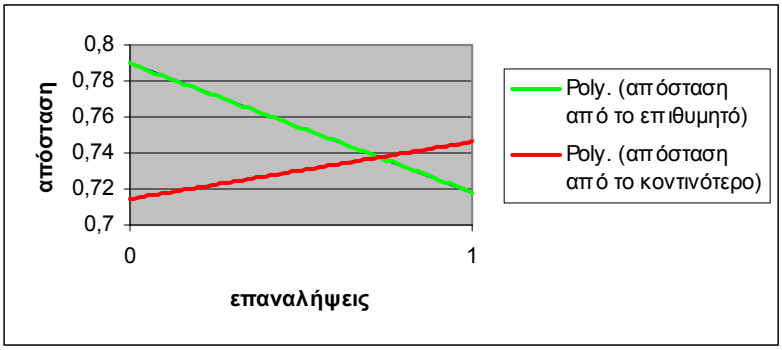
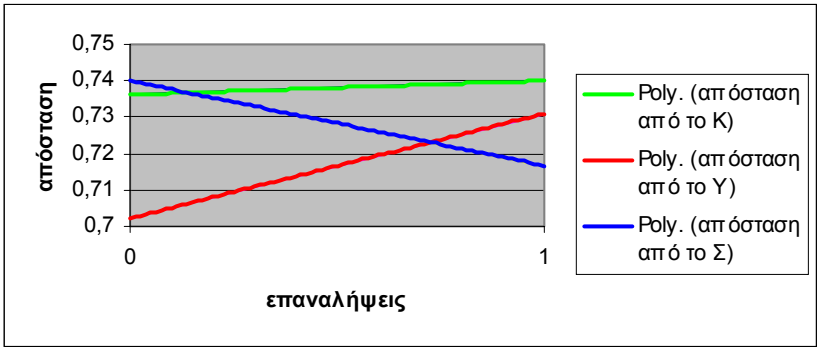
Αρχικά το κοντινότερο μοντέλο ήταν το Ψ:

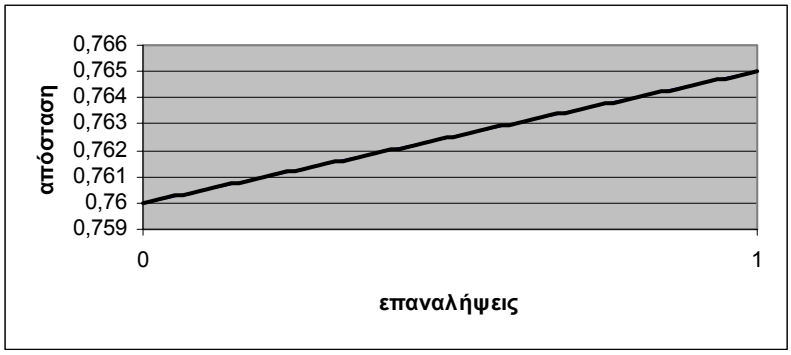
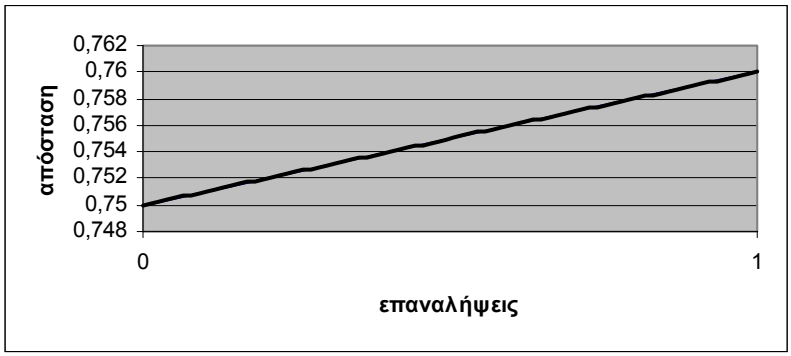
Ενώ με την πρώτη αύξηση το κοντινότερο μοντέλο ήταν το Υ:

4.5.2 Εφαρμογή του αλγορίθμου με διαφορετικό κόψιμο της δεύτερης εικόνας

Εφαρμόσαμε τον αλγόριθμο στην δεύτερη εικόνα με διαφορετικό κόψιμο αυτή την φορά όπως φαίνεται παρακάτω. Στον πίνακα που ακολουθεί φαίνονται τα αποτελέσματα.



Αρχική εικόνα	**Κοντινότερο γράμμα**	**Μεταβολή των αποστάσεων ομοιότητας από το κοντινότερο και από το επιθυμητό γράμμα**
1. 	**Η** 	
2. 	**Β** 	
3. 	**Χ** 	
4. 	**Υ** 	

5. 	**I** 	
6. 	**T** 	

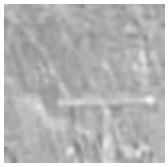
4.5.3 Σχολιασμός των αποτελεσμάτων

Περίπτωση 1:

Στο πρώτο classificati□n κοντινότερος χαρακτήρας ήταν το Η:

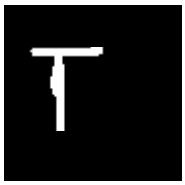
Με την πρώτη αύξηση του Thresh□ld το κομμάτι θορύβου και ένα μέρος του γράμματος αποκολήθηκαν από τον χαρακτήρα και κοντινότερο μοντέλο ήταν το Γ:

Περίπτωση 3:



Αρχικά ο χαρακτήρας που έμοιαζε περισσότερο στο Τ της εικόνας μας ήταν το Χ:

Με την αύξηση του thresh□ld , το κομάτι θορύβου αποκολήθηκε και έτσι το classificati□n έγινε σωστά, δίνοντάς μας κοντινότερο χαρακτήρα το Τ:



Περίπτωση 4:

Εδώ αρχικά κοντινότερος χαρακτήρας ήταν το Υ:

Ενώ η δεύτερη επανάληψη έκοψε ένα κομμάτι του χαρακτήρα και έδωσε ως κοντινότερο γράμμα το Σ:

Περίπτωση 6:

Σε αυτήν την εικόνα εμφανίζονται δύο χαρακτήρες, το Σ και το Ι.στην εικόνα που προκύπτει μετά την επεξεργασία το Σ χάθηκε και έμεινε το Ι το οποίο και αναγνωρίστηκε σωστά.

Τέλος πρέπει να σημειωθεί ότι και σε αυτές τις εικόνες η επεξεργασία έγινε με την μέθοδο του Fitting για την εύρεση του Thresh□ld.

4.6 Γενικά σχόλια, συμπεράσματα και επεκτάσεις

Από τα παραπάνω αποτελέσματα υπολογίσαμε ότι η μέθοδός μας είχε περίπου 50% ποσοστό επιτυχίας. Εξετάζοντας τα στάδια από την αρχή μπορούμε να εντοπίσουμε μειονεκτήματα και προτερήματα της μεθόδου μας.

Το αρχικό Segmentati□n της εικόνας μας σε περίπτωση που τα γράμματα είναι διαφορετικά κατανεμημένα ίσως να αποτύχει να κόψει τις περιοχές με τέτοιον τρόπο ώστε να παίρνουμε όλον τον χαρακτήρα. Έτσι θα μπορούσαμε να εφαρμόσουμε στην αρχική εικόνα μας τον αλγόριθμο Minimum Spanning Tree που θα μας έδινε ως γειτονικές περιοχές εικόνες που περιέχουν γράμμα. Αυτός ο τρόπος θα δούλευε ακόμα καλύτερα σε συνδιασμό με προηγούμενη δουλειά που έχει γίνει και απορίπτει περιοχές θορύβου.

Η τεχνική για την βελτίωση του Thresh□ld είχε καλά αποτελέσματα αφού στις περιπτώσεις όπου το γράμμα ήταν ευδιάκριτο πλησιάζαμε πολύ το ιδανικό Thresh□ld. Στις περιπτώσεις όπου αποτύχαμε θα μπορούσαμε να βελτιώσουμε τα αποτελέσματα διαλέγοντας διαφορετική κατανομή για να μοντελοποιήσουμε την περιοχή του γράμματος και να υπολογίσουμε ένα καλύτερο thresh□ld.

Οι m□rph□l□gical □perati□ns σε συνδιασμό με τον αλγόριθμο Minimum Spanning Tree για τον εντοπισμό και την εξαγωγή του χαρακτήρα δούλεψαν αρκετά ικανοποιητικά αφού η έντονη παρουσία θορύβου έκανε την δουλειά μας αρκετά δύσκολη.

Τέλος ο αλγόριθμος Ge□metric M□ments έδειξε ότι το σύστημά μας είναι αρκετά ασταθές αφού μικρές αλλαγές στην τιμή του Thresh□ld αλλάζουν πολύ τα τελικά αποτελέσματα και επηρεάζουν το Classificati□n. Δεν έχει όμως αναφερθεί άλλη μέθοδος

στην βιβλιογραφεία ειδικά για αναγνώριση γραφής σε μαρμάρινες επιγραφές για αυτό και η απόπειρα αυτή είναι σε αρκετά πρώιμο στάδιο και επιδέχεται αρκετές αλλαγές και δοκιμή άλλων μεθόδων ίσως πιο σταθερών.

References

[1] The Image □r□cessing □andb□□k, 3rd editi□n J□hn C. Russ

[2] Digital Image □r□cessing Rafael C. G□nzalez and Richard □. W□□ds.

[3] □drian □. Whichell□and □□ng □an ‘Linking br□ken character b□rders with variable sized masks t□impr□ve rec□gniti□n’, □attern rec□gniti□n, V□l. 29, N□ 8, pp. 1429-1435, 1996.

[4] □abu M., Methtre, M□han S. □ankanhalli and Wing F□□n Lee ‘Shape measures f□r c□ntent-based image retrieval: □ c□mparis□n’, Inf□rmati□n □r□cessing & Management, V□l. 33, N□3, pp. 319-337, 1997.

[5] Ming-□uei □u, ‘Visual □attern Rec□gniti□n by m□ment invariants’, IR□ transacti□ns □n inf□rmati□n the□ry.

[6] Ιωάννης Πήτας, ‘Ψηφιακή επεξεργασία εικόνας’.