

Ονοματεπώνυμο : Αριστείδης Πηλιανίδης

ΑΕΜ : 8755

Ψηφιακή επεξεργασία εικόνας

Άσκηση 3

Demo1:

clust2 clust3 clust4

1 1 4

1 1 2

1 1 4

1 1 4

1 2 3

1 2 3

1 2 3

1 2 3

2 3 1

2 3 1

2 3 1

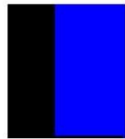
2 3 1

time2	time3	time4
2.3262	0.0721	0.0225

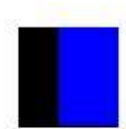
Demo2:

Για την 1^η εικόνα:

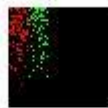
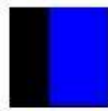
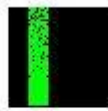
K=2:



K=3:



K=4:



Για την 2^η εικόνα:

K=2:



K=3:



K=4:

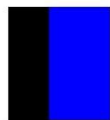


Το συμπέρασμα που βγάζουμε είναι ότι ο Spectral clustering δουλεύει καλύτερα σε εικόνες που έχουν περιοχές με χρώμα, όσες είναι και οι περιοχές που θέλουμε να χωρίσουμε την εικόνα.

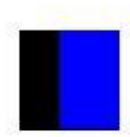
Demo3a:

Για την 1^η εικόνα:

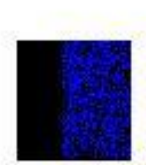
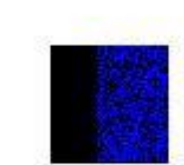
K=2:



K=3:



K=4:



Για την 2^η εικόνα:

K=2:



K=3:



K=4:



Συμπεράσματα για το demo3a:

- 1) τα αποτελέσματα που παίρνουμε δεν είναι ίδια στους αλγορίθμους Spectral-Ncuts.
- 2) ο Ncuts κάνει καλύτερο clustering γιατί πετυχαίνει μεγαλύτερη συσχέτιση στο εσωτερικό των περιοχών που χωρίζει καθώς και μεγαλύτερη ασυσχέτιστη μεταξύ των διαφορετικών περιοχών.
- 3) Ο Ncuts είναι σε γενικές γραμμές πιο αργός από τον Spectral (λογικό εφόσον μας δίνει και καλύτερα αποτελέσματα) αν και βλέπουμε ότι για $k=4$ ο Ncuts είναι περίπου 2 φορές γρηγορότερος (επαληθεύτηκε με πολλές δοκιμές στο matlab).

Οι χρόνοι για το spectral:

1^ο πείραμα

time2 time3 time4

3.5995 4.2736 3.2727 (sec)

2^ο πείραμα

time2 time3 time4

5.0991 6.3784 37.6097 (sec)

Οι χρόνοι για το Ncuts:

1^ο πείραμα

time2 time3 time4

19.6265 17.9583 16.6629

2^ο πείραμα

time2 time3 time4

18.1256 20.7634 19.0249

Συμπεράσματα για το demo3b:

Αρχικά να πούμε πως για την αναδρομική μέθοδο για **ένα βήμα** περιμένουμε ίδια αποτελέσματα με την μη-αναδρομική μέθοδο για $k=2$ για την οποία τα αποτελέσματα συγκριτικά με τον spectral clustering τα βλέπουμε στα συμπεράσματα για το demo3a ακριβώς πιο πάνω.

Διευκrynήσεις:

- 1) Το δεύτερο πείραμα που αφορά την εικόνα d2b βρίσκεται στο demo3b2.
- 2) Το όριο T2 παρόλλο που βρήκα τις τιμες nCutValue δεν το χρησιμοποίησα για να σταματήσω την αναδρομή γιατί είναι το ίδιο πράγμα με το όριο T1.
- 3) Η λογική που ακολουθησα είναι η εξης:
Αρχικα το πρωτο βημα της αναδρομικης διαδικασιας γινεται εκτος αναδρομης , διότι για να δουλεψει η αναδρομή, χρειαζομαι τους πίνακες WA και WB του πατερα, οποτε εφόσον δεν υπάρχει πατέρας στο πρώτο βήμα δεν μπορεί να γινει μεσα στην αναδρομη.

Η μεταβλητη count μετραει σε πιο κομβο του δεντρου ειμαστε, ξεκινωντας να μετραμε από αριστερα προς τα δεξιά.

Το οριο T1 το θέτουμε αρκετα υψηλα ώστε να είναι μεγαλύτερο από τον αριθμο των περιοχων που χωρίζεται η πρώτη εικονα με αποτελεσμα να εκτελείται ένα βημα του αλγοριθμου.

Η μεταβλητή `sth` χρησιμοποιείται στην συνάρτηση `myRecNcutsB` μόνο, και χρησιμευει ώστε να ελεγχουμε κάθε φορά τον πατέρα του κόμβου στον οποίο βρισκόμαστε. Κατά τα άλλα οι `myRecNcutsB` και `myRecNcutsA` είναι ίδιες.

Το `level` είναι μεταβλητή που μας δείχνει σε πιο επιπεδο του δεντρου βρισκόμαστε, ώστε να καλούμε οσες φορές χρειάζεται κάθε φορά τις συναρτησεις `myRecNcutsB` και `myRecNcutsA`.

Το `c` είναι απλα ένας index για τον πινακα `filo` στον οποιο αποθηκευουμε τους κομβους που εχει γινει τουλαχιστον μια διχοτομηση, οποτε πρεπει να δειξουμε τις εικόνες που προέκυψαν από αυτήν την διχοτόμηση.

Στην συνέχεια, βρίσκουμε ότι χρειαζομαστε και τα αποθηκευουμε σε struct array. Ελεγχουμε αν ο αριθμος των κομβων που αντιστοιχουν στα cluster 1 και 2 είναι μεγαλύτερος από το T1 οποτε και απόθηκευουμε τους affinity πινακες που βρικαμε πιο πανω, διοτι θα μας χρειασουν. Αν είναι μικροτερος από T1, τοτε για memory efficiency βαζουμε στους αντιστιχους πινακες τον empty πινακα. Τέλος αποθηκεύουμε στον πινακα `filo` τα “φυλλα” μας, που είναι οι κόμβοι στους οποίους εχει γινει τουλάχιστον μια διχοτόμηση.

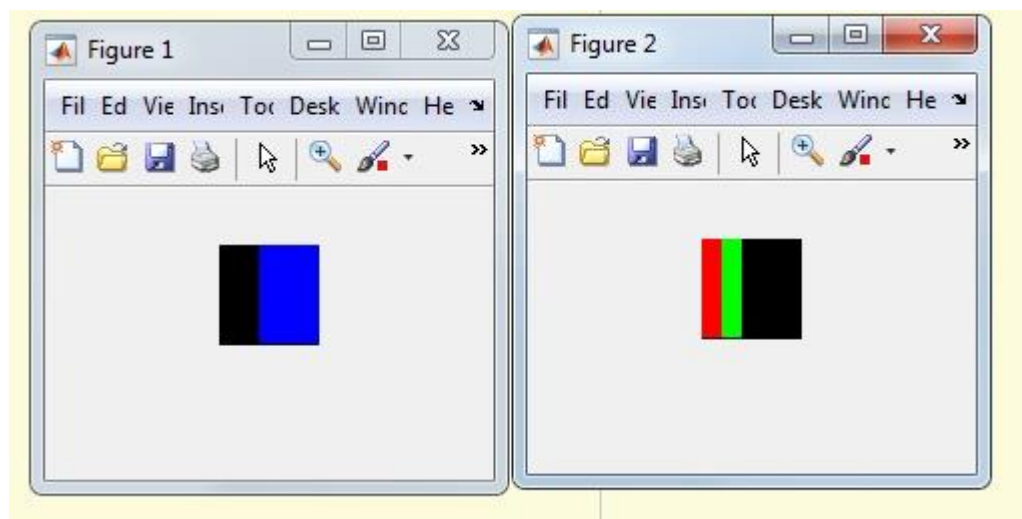
Μέσα στην αναδρομή τώρα, καλούμε όσες φορές πρέπει ανάλογα το level, τις συναρτήσεις myRecNcutsB και myRecNcutsA. Σε αυτές ελέγχουμε αν ο πατέρας του κόμβου στον οποίο βρισκόμαστε ήταν ο empty πίνακας, οπότε και αυξάνουμε τον teloscounter κατά 1 που μας δείχνει αν είναι ίσος με το $(2^{\text{level}+1})$, ότι έχει τελειώσει η αναδρομή. Αν δεν ήταν ο empty πίνακας σημαίνει ότι συνεχίζει η αναδρομή από αυτόν τον κόμβο οπότε επαναλαμβάνουμε την παραπάνω διαδικασία.

Τώρα λόγω error του matlab:

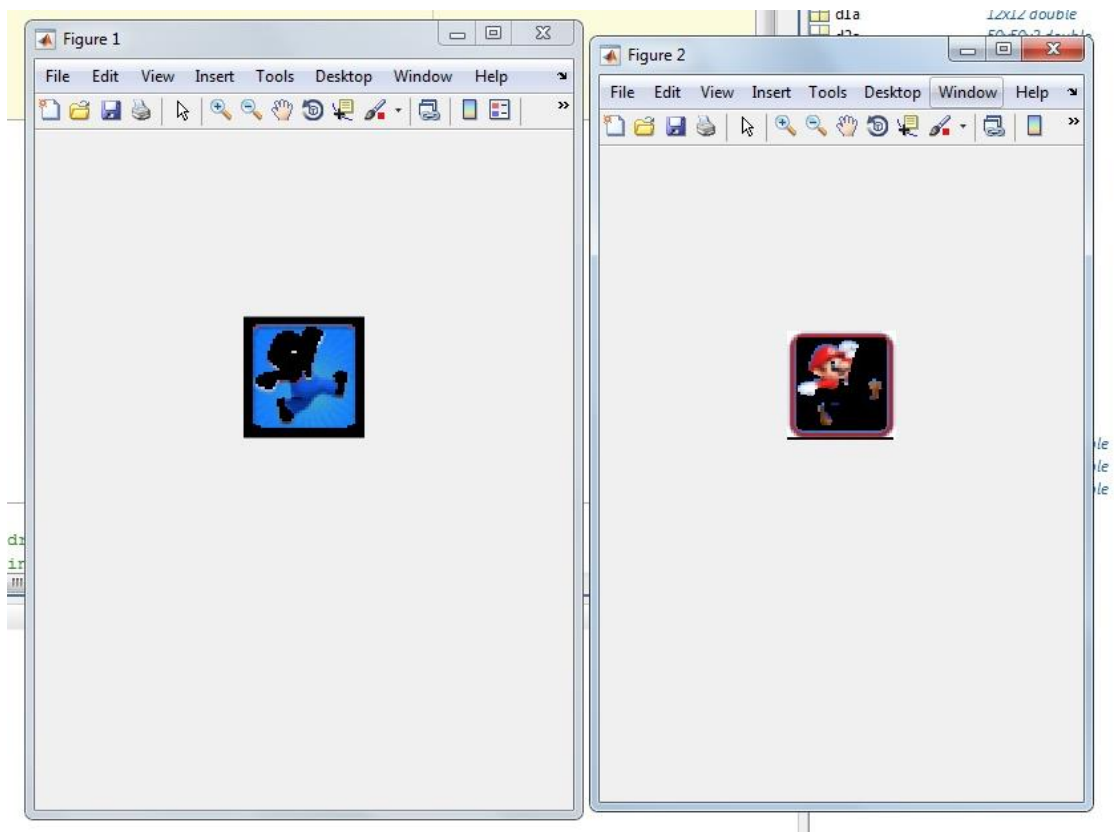


Που δεν ξέρω γιατί προέκυψε, από εδώ και πέρα οι εικόνες παρουσιάζονται μέσω print Screen.

Demo3b(d2a):



Demo3b(d2b):



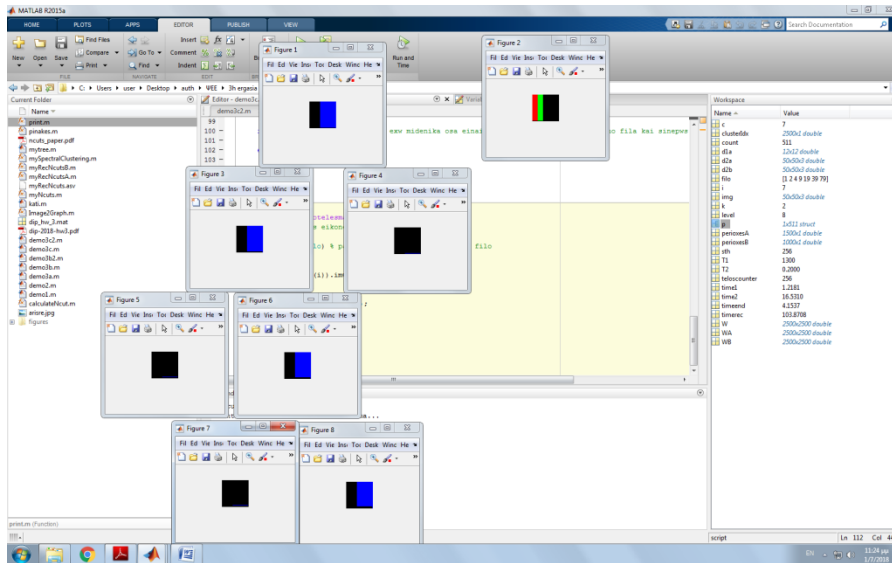
Συνεπώς βλέπουμε ότι όντως έχουμε τα ίδια αποτελέσματα με την μη-αναδρομική διαδικασία για $k=2$.

Demo3c:

Διευκrynήσεις:

- 1) Το δεύτερο πείραμα που αφορά την εικόνα d2b βρίσκεται στο demo3c2.
- 2) Το όριο T2 παρόλλο που βρήκα τις τιμες nCutValue δεν το χρησιμοποίησα για να σταματήσω την αναδρομή γιατί είναι το ίδιο πράγμα με το όριο T1.
- 3) Η λογική είναι ίδια με το demo3b, απλα έχουμε μικρότερο T1(ή T2 αντιστοιχα)

Demo3c(d2a):



Εχω παραλείψει κάποια figures γιατί δεν χωρούσαν στην εικόνα. Εδώ αν γίνει zoom στην εικόνα θα παρατηρήσουμε ότι κάθε φορά που γίνεται διχοτόμηση στην μπλέ εικόνα(γιατι αυτή ικανοποιεί τον περιορισμό για το T1) η εικόνα χωρίζεται σε μια μικρη μπλε λωριδα(εικόνα με μαυρο φοντο) και στην υπολοιπη μπλε εικόνα.

Demo3c(d2b):

