

# Ψηφιακή Επεξεργασία Εικόνας

## Εργασία 2

Βασίλειος Αραϊλόπουλος  
varailop@ece.auth.gr  
AEM: 9475

Μάιος 2021

### 1 Εικόνες ως γράφοι

Ζητούμενο αυτού του ερωτήματος ήταν η κατασκευή ενός πίνακα που περιγράφει τον γράφο των pixel της εικόνας, που τα βάρη των ακμών του είναι:

$$A(i, j) = \frac{1}{e^{d(i, j)}},$$

όπου  $d(i, j)$  η Ευκλείδεια απόσταση μεταξύ των τιμών του pixel. Στην υλοποίηση του κώδικα εκτελείται μια διπλή for loop που γεμίζει τα στοιχεία του κάτω τριγωνικού πίνακα affinity με τα κατάλληλα βάρη  $A(i, j)$  και στην συνέχεια εκτελείται η πράξη:

$$Affinity + Affinity^T - diag(Affinity)$$

για να δημιουργηθεί ο συμμετρικός πίνακας. Τα στοιχεία είναι αποθηκευμένα στον πίνακα με row-major order, δηλαδή για να αναπαρασταθούν τα  $M \times N$  pixel στην μία διάσταση του affinity πίνακα (μέγεθος  $(MN) \times (MN)$ ) «διαβάζονται» από την εικόνα τα  $N$  στοιχεία από μια γραμμή και μετά προχωράει στην επόμενη κοκ.

### 2 Spectral Clustering

Στην ενότητα αυτή η συνάρτηση παίρνει ως δεδομένο τον affinity πίνακα μιας εικόνας όπως προκύπτει από το ερώτημα 1 και τον χωρίζει σε clusters. Για την υλοποίηση της συνάρτησης, αρχικά, υπολογίζεται ο  $D$  διαγώνιος πίνακας με  $D(i, i) = \sum_j Affinity(i, j)$ . Μετά, υπολογίζονται για τον πίνακα  $L = D - Affinity$  οι  $k$  μικρότερες ιδιοτιμές του, χρησιμοποιώντας στην συνάρτηση *eigs* την επιλογή 'smallestreal' και τέλος καλείται η συνάρτηση *kmeans* για τον πίνακα ιδιοδιανυσμάτων  $U$ .

## 2.1 Demo 1

Παρακάτω φαίνονται τα αποτελέσματα που προκύπτουν για τον έτοιμο affinity πίνακα *d1a* με διαχωρισμό σε 2, 3 και 4 clusters.

```
For k = 2 the cluster is: 2 2 2 2 1 1 1 1 2 2 2 2
For k = 3 the cluster is: 1 1 1 1 2 2 2 2 3 3 3 3
For k = 4 the cluster is: 1 1 1 1 2 3 3 3 4 4 4 4
```

Figure 1: Οι ετικέτες των κόμβων του *d1a* για  $k = 2, 3, 4$ .

Η συνάρτηση φαίνεται να έχει χωρίσει τον πίνακα του γράφου με σωστό τρόπο, αφού σε κάθε περίπτωση περιέχονται όλες οι ετικέτες από το 1 έως το  $k$ . Στην πρώτη περίπτωση έχουμε τους κόμβους χωρισμένους σε 3 τετράδες με την πρώτη και την τελευταία να περιέχονται στο ίδιο cluster. Στην δεύτερη περίπτωση κάθε μία από τις προηγούμενες τετράδες ανήκει σε διαφορετικό cluster. Και στην τελευταία περίπτωση σε σύγκριση με την δεύτερη φαίνεται να έχει χωριστεί η δεύτερη τετράδα σε δύο μέρη. Ωστόσο, από αυτές τις περιπτώσεις είναι δύσκολο να επιλέξουμε την καταλληλότερη λύση στο πρόβλημα μας.

## 2.2 Demo 2

Για το demo 2 έπρεπε να χρησιμοποιηθεί η συνάρτηση του ερωτήματος 1 και να μετατραπούν οι εικόνες σε γράφους. Μετά έπρεπε να εφαρμοστούν αυτοί οι γράφοι στην συνάρτηση *mySpectralClustering* και να βρεθούν τα clusters για  $k = 2, 3$  και 4. Τέλος, με την βοήθεια της μεθόδου *reshape* μετατράπηκε η row-major στήλη στον αρχικό  $M \times N$  πίνακα και διαιρέθηκαν τα στοιχεία των clusters με την μέγιστη τιμή που υπήρχε για να λάβει η εικόνα τιμές από το 0 έως το 1. Τα αποτελέσματα του demo 2 είναι:

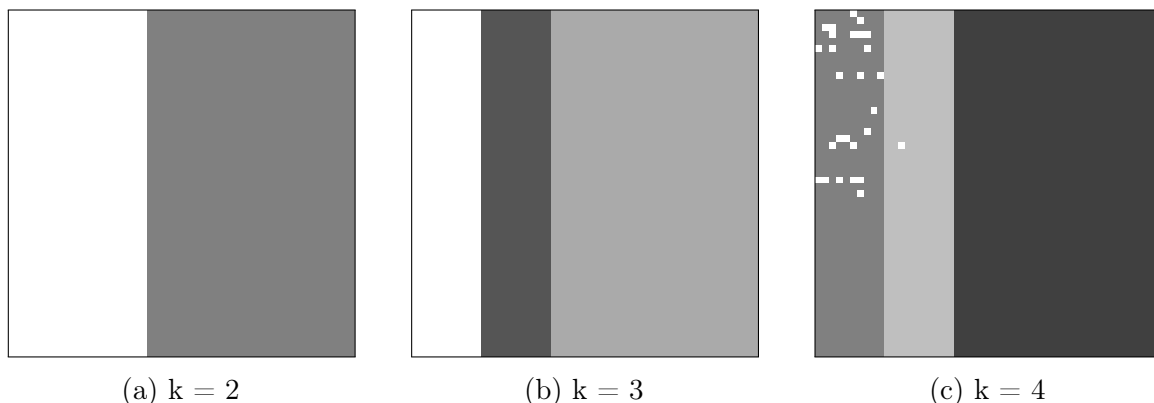


Figure 2: Spectral Clustering εφαρμοσμένο στην εικόνα *d2a* για  $k = 2, 3, 4$ .

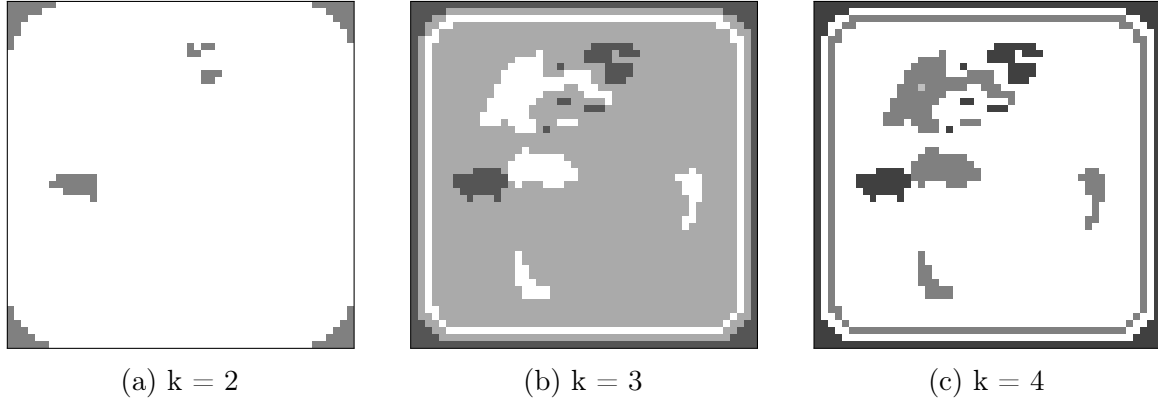


Figure 3: Spectral Clustering εφαρμοσμένο στην εικόνα d2b για  $k = 2, 3, 4$ .

Παρατηρώντας την εικόνα d2a είναι οπτικά ξεκάθαρο ότι ο αλγόριθμος δουλεύει πολύ σωστά για  $k = 3$  αφού διαχωρίζει πλήρως τα τρία χρώματα της εικόνας. Ακόμα και για την περίπτωση που  $k = 2$  δίνει καλύτερα αποτελέσματα από την  $k = 4$ , γιατί ο διαχωρισμός των clusters σημαίνει διαφορετικά χρώματα, κάτι που κάνει στην περίπτωση 2c να θεωρηθεί «άκυρη» επειδή υπάρχουν διαφορές στην μεσαία λωρίδα. Στην εικόνα d2b αρχικά διαχωρίζονται τα χέρια του Mario και περιοχές έξω από το περίγραμμα του εικονιδίου. Από την δεύτερη εικόνα και μετά αρχίζουν και φαίνονται καλύτερα τα χαρακτηριστικά του, αλλά τα πόδια και το σώμα του δεν έχουν διαχωριστεί από το background και στην τελευταία εικόνα δεν υπάρχει σχεδόν καμία διαφοροποίηση πέρα από ένα μοναδικό pixel στο καπέλο του που έχει διαφορετική απόχρωση. Άρα καταλαβαίνουμε ότι ούτε εδώ έχουμε πολύ καλό διαχωρισμό της εικόνας.

### 3 Normalized-cuts

Παρόμοια με την μέθοδο που περιγράφηκε στην ενότητα 2 με την διαφορά ότι πρέπει να λυθεί το γενικευμένο πρόβλημα ιδιοτιμών  $Lx = \lambda Dx$ . Επίσης, χρειάστηκε να υλοποιηθεί και μια συνάρτηση που υπολογίζει την τιμή  $ncut$  όπως περιγράφεται στην εκφώνηση. Σε αυτήν την συνάρτηση διατρέχονται όλα τα στοιχεία του affinity πίνακα και ανάλογα με τα cluster στα οποία ανήκουν τα δύο pixel προστίθεται το στοιχείο του πίνακα στην κατάλληλη assoc. Μόλις βρεθούν οι  $assoc(A,A)$ ,  $assoc(A,V)$ ,  $assoc(B,B)$  και  $assoc(B,V)$ , υπολογίζεται το  $Nassoc(A,B)$  και συνεπώς η ζητούμενη τιμή  $ncut$ .

#### 3.1 Demo 3a

Παρόμοιο demo με το 2, μόνο που στην θέση του *mySpectralClustering* χρησιμοποιείται η *myNCuts*. Παρακάτω φαίνονται τα αποτελέσματα αυτής της μεθόδου.

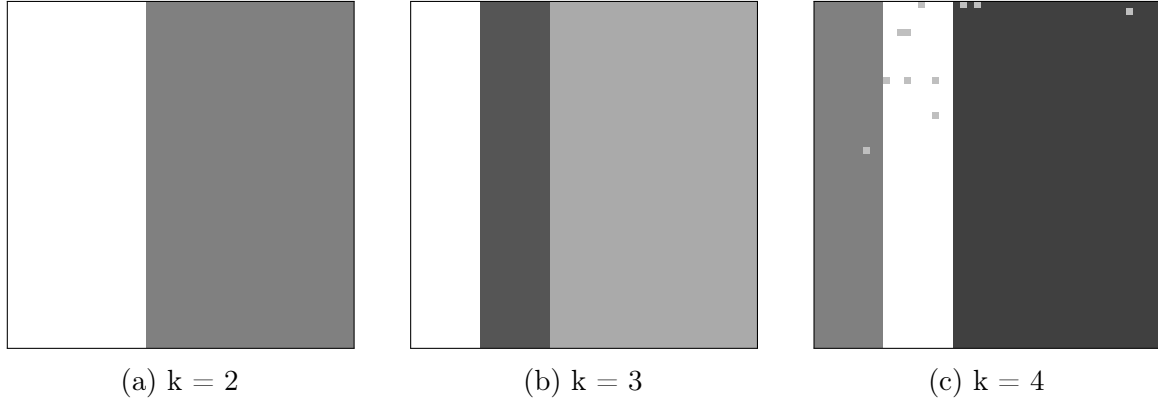


Figure 4: Ncut εφαρμοσμένο στην εικόνα d2a για  $k = 2, 3, 4$ .

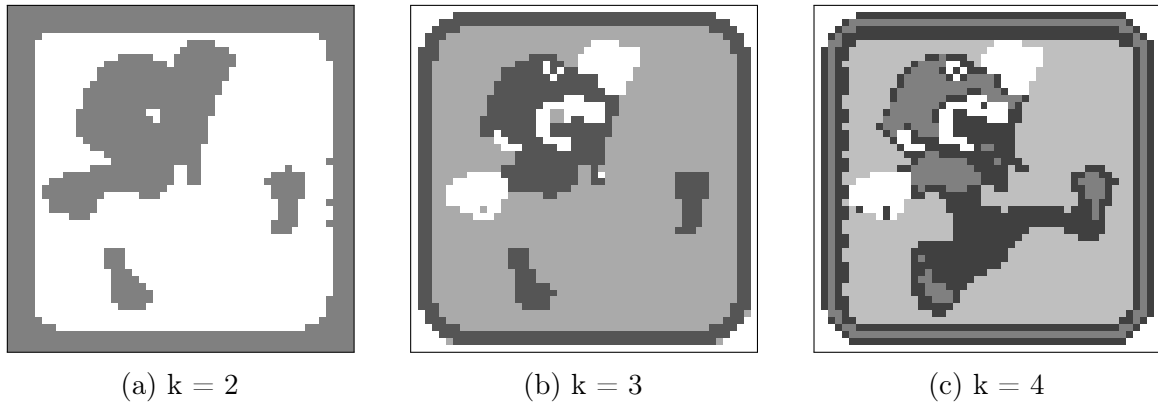


Figure 5: Ncut εφαρμοσμένο στην εικόνα d2b για  $k = 2, 3, 4$ .

Σε σύγκριση με την προηγούμενη μέθοδο, στην περίπτωση του σχήματος 4 δεν βλέπουμε σημαντικές διαφορές, πέρα από την τελευταία εικόνα. Ωστόσο, στην περίπτωση του σχήματος 5 έχουμε αρκετές διαφορές. Στην 5a βλέπουμε να έχει γίνει ένας σχετικά καλύτερος διαχωρισμός της εικόνας σε σχέση με αυτή του 3a. Δεν έχουν διαχωριστεί μόνο τα χέρια του Mario, αλλά και το κεφάλι και τα παπούτσια του. Δεν έχει γίνει βέβαια ο πλήρης διαχωρισμός της φιγούρας του, όμως είναι οπτικά καλύτερο από το προηγούμενο. Ακόμα και η 5b είναι πιο καλή από την 3b γιατί αρχίζουν να σχηματίζονται τα χαρακτηριστικά του προσώπου και οι λεπτομέρειες στο καπέλο του. Τέλος, στην 5c γίνονται πιο έντονα τα χαρακτηριστικά του και αρχίζει να διακρίνεται και μέρος των ποδιών του από το background, που στην αρχική εικόνα οπτικά φαίνονται τα χρώματα να είναι κοντά.

### 3.2 Demo 3b

Ίδιος κώδικας με το demo 3a που εκτελείται μόνο για την περίπτωση  $k = 2$  και υπολογίζονται οι μετρικές τιμές ncut για τα 2 clusters που προκύπτουν. Οι εικόνες που προκύπτουν από την εκτέλεση του κώδικα είναι οι 4a και 5a και οι τιμές που υπολογίζονται είναι:

The n-cut value for d2a is: 0.509238 and for d2b is: 0.785285

Figure 6: Οι μετρικές των τιμών nCut για τις εικόνες d2a και d2b μετά τον πρώτο διαχωρισμό.

### 3.3 Demo 3c

Για τον σκοπό αυτού του demo δημιουργήθηκε μια επιπλέον συνάρτηση με όνομα *myNCutsRecursive*, η οποία υλοποιεί την αναδρομική μέθοδο της normalized-cuts. Δέχεται ως ορίσματα τον affinity πίνακα, τις τιμές  $T^1$ ,  $T^2$  και μια τιμή first που λαμβάνει την τιμή 1 όταν εκτελείται για πρώτη φορά η μέθοδος και την τιμή 0 αν εκτελεστεί από την αναδρομή. Αρχικά, χωρίζεται ο πίνακας με την συνάρτηση *myNCuts* και ελέγχονται οι περιορισμοί, αν δηλαδή ο αριθμός των κόμβων με μια ετικέτα είναι μικρότερος από το  $T^1$  ή αν η τιμή  $ncut$  (που υπολογίζεται από την συνάρτηση *calculateNcut*) είναι μεγαλύτερη από το  $T^2$ . Σε αυτές τις δύο περιπτώσεις ο αλγόριθμος σταματά και αν είναι η πρώτη κλήση της συνάρτησης επιστρέφεται το *clusterIdx*, αλλιώς επιστρέφει πίνακα με άσσους ίσου μεγέθους με το *clusterIdx*. Αν, όμως, συνεχιστεί η αναδρομή, τότε χωρίζεται σε δύο μέρη ο γράφος, δηλαδή βρίσκονται δύο καινούριοι affinity πίνακες και αποθηκεύονται τα indices από τα pixel. Εκτελείται έπειτα για κάθε μέρος αναδρομικά η συνάρτηση της μεθόδου και επιστρέφονται τα δύο νέα *clusterIdx*. Προστίθεται η μέγιστη ετικέτα του ενός *clusterIdx* στο άλλο για να μην είναι ίδιες οι ετικέτες και τέλος σύμφωνα με τα indices που έχουν αποθηκευτεί τοποθετούνται στην κατάλληλη θέση και επιστρέφεται το νέο *clusterIdx*. Για περισσότερη κατανόηση υπάρχουν βοηθητικά σχόλια στο αρχείο της συνάρτησης.

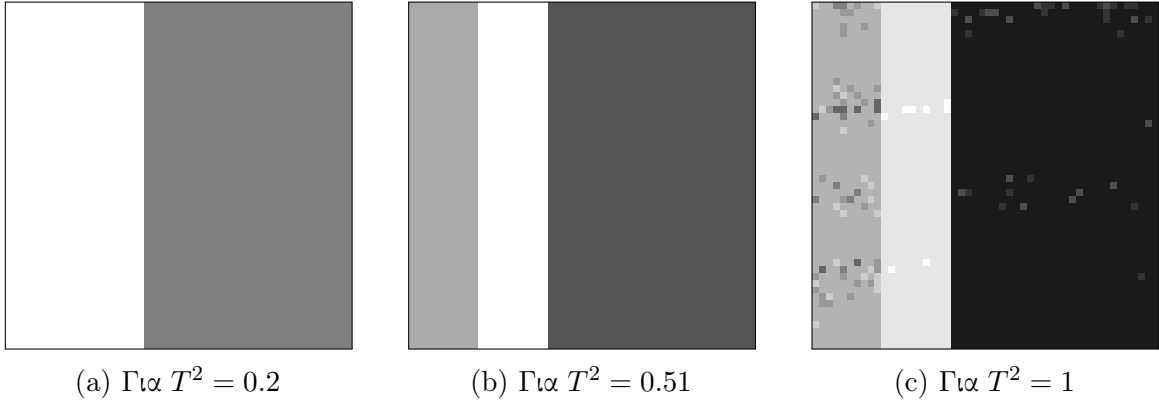


Figure 7: Ncut αναδρομικό εφαρμοσμένο στην εικόνα d2a για  $T^2 = 0.2, 0.51, 1$ .

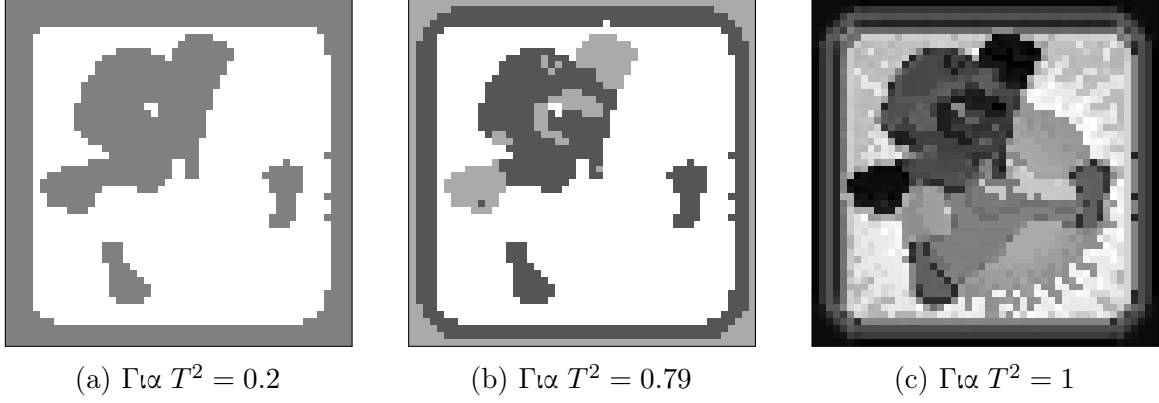


Figure 8: Neut αναδρομικό εφαρμοσμένο στην εικόνα d2b για  $T^2 = 0.2, 0.51, 1$ .

Εφαρμόζοντας τον αναδρομικό αλγόριθμο για  $T^1 = 5$  και  $T^2 = 0.2$  παίρνουμε για τις δύο περιπτώσεις τις εικόνες που φαίνονται στις εικόνες 7a και 8a. Όπως είναι λογικό επειδή στην εικόνα d2a έχουμε από το προηγούμενο ερώτημα ότι η τιμή της neut είναι 0.5 και για την εικόνα d2b είναι 0.78 και αυτά είναι μεγαλύτερα από το  $T^2 = 0.2$  δεν θα εκτελεστεί καμία αναδρομή και το  $k$  θα είναι 2. Ωστόσο, αν αυξηθεί η τιμή του  $T^2$  λίγο πάνω από τις τιμές αυτές του neut για κάθε εικόνα θα προκύψουν τα αποτελέσματα που φαίνονται στις εικόνες 7b και 8b. Στην 7b το  $k$  γίνεται 3 (που είναι και το πιο σωστό) και στην 8b έχουμε  $k = 3$ . Στις εικόνες 7c και 8c έχουμε μια ακραία περίπτωση που το  $T^2$  παίρνει την τιμή 1 και στην πρώτη περίπτωση έχουμε  $k = 10$  και στην δεύτερη  $k = 208$ .