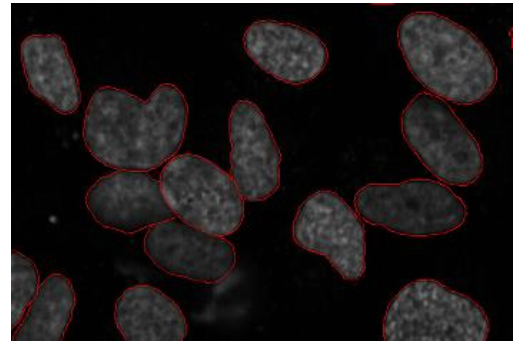
ΤΕΧΝΗΤΗ ΟΡΑΣΗ ΤΠ70Υ2

Διάλεξη 8

Κώστας Μαριάς
Αναπληρωτής Καθηγητής Επεξεργασίας Εικόνας

I am not eternity, but a man; a part of the whole,
as an hour is of the day.

- Epictetus Ancient Greece Philisopher



Κατάτμηση Εικόνας - Image Segmentation

Κατάτμηση Εικόνας - Image Segmentation

- Η τμηματοποίηση υποδιαιρεί μια εικόνα σε διαχωρίσιμες περιοχές ή τα αντικείμενα της.
- Το επίπεδο στο οποίο μεταφέρεται η υποδιαίρεση εξαρτάται από το πρόβλημα, δηλαδή, η κατάτμηση πρέπει να σταματήσει όταν τα αντικείμενα που ενδιαφέρουν έχουν απομονωθεί.
- Για παράδειγμα, στην αυτοματοποιημένη επιθεώρηση ηλεκτρονικών, το ενδιαφέρον έγκειται στην ανάλυση των εικόνων των προϊόντων με στόχο να εντοπιστούν συγκεκριμένες ανωμαλίες, όπως αν λείπουν εξαρτήματα ή αν υπάρχουν διακεκομμένες διαδρομές σύνδεσης.

Κατάτμηση Εικόνας - Image Segmentation

- Σε ορισμένες περιπτώσεις, όπως οι εφαρμογές βιομηχανικής επιθεώρησης, τουλάχιστον σε κάποιο βαθμό ο έλεγχος του περιβάλλοντος απεικόνισης είναι δυνατός κατά καιρούς.
- Σε άλλες, όπως στην τηλεπισκόπηση, ο έλεγχος του χρήστη κατά την απόκτηση της εικόνας περιορίζεται κυρίως στην επιλογή του αισθητήρα απεικόνισης.

Κατάτμηση Εικόνας - Image Segmentation

https://en.wikipedia.org/wiki/Image_segmentation

- Ο στόχος της τμηματοποίησης είναι να απλοποιήσει την αναπαράσταση μιας εικόνας σε κάτι που είναι πιο ουσιαστικό και ευκολότερο να αναλυθεί.
- Σε πιο πρακτικό επίπεδο, η κατάτμηση εικόνων χρησιμοποιείται συνήθως για τον εντοπισμό αντικειμένων και ορίων (γραμμών, καμπυλών κ.λπ.) στις εικόνες.
- Σε μια πιο ευρεία θεώρηση, η κατάτμηση της εικόνας είναι η διαδικασία εκχώρησης μιας ετικέτας σε κάθε εικονοστοιχείο σε μια εικόνα, έτσι ώστε εικονοστοιχεία με την ίδια ετικέτα να μοιράζονται χαρακτηριστικά που ενδιαφέρουν τον χρήστη.

Κατάτμηση Εικόνας - Image Segmentation

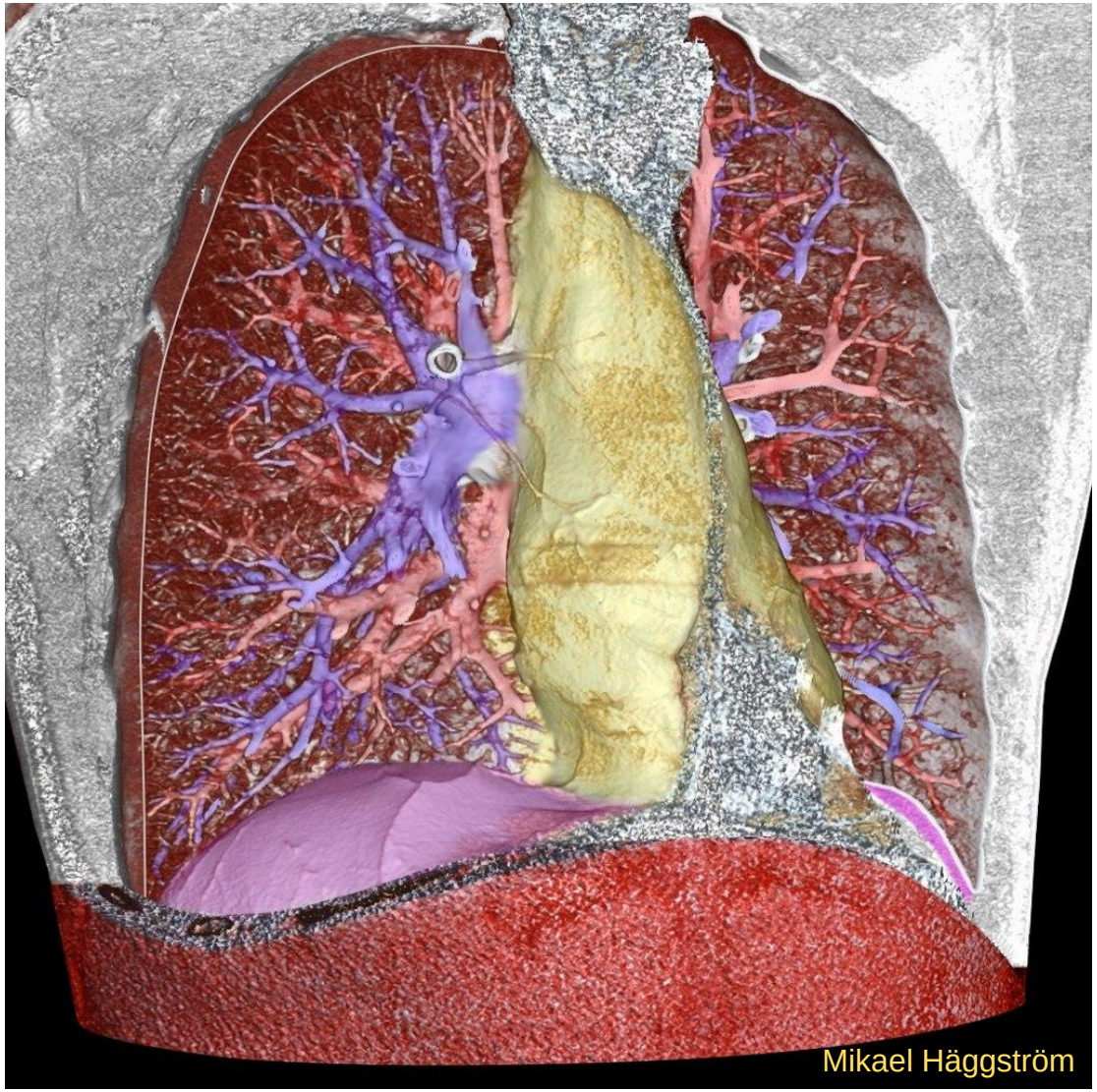
https://en.wikipedia.org/wiki/Image_segmentation

- Το αποτέλεσμα της τμηματοποίησης εικόνας είναι ένα σύνολο από τμήματα που καλύπτουν ολόκληρη την εικόνα ή ένα σύνολο από περιγράμματα που εξάγονται από την εικόνα (ανίχνευση ακμών).
- Κάθε ένα από τα εικονοστοιχεία κάθε περιοχής μοιάζουν σε κάποια χαρακτηριστική ή υπολογιζόμενη ιδιότητα, όπως χρώμα, ένταση ή υφή.
- Ανάλογα, γειτονικές περιοχές διαφέρουν σημαντικά σε σχέση με τα ίδια χαρακτηριστικά.
- Η τμηματοποίηση μπορεί να εφαρμοστεί και σε 3Δ δεδομένα (π.χ. ιατρική απεικόνιση).

Κατάτμηση Εικόνας - Image Segmentation

https://en.wikipedia.org/wiki/Image_segmentation

Τμηματοποίηση 3D του θώρακα: Το πρόσθιο θωρακικό τοίχωμα, οι αεραγωγοί και τα πνευμονικά αγγεία μπροστά στη ρίζα του πνεύμονα αφαιρέθηκαν ψηφιακά για να απεικονίσουν το θωρακικό περιεχόμενο: - μπλε: πνευμονικές αρτηρίες - κόκκινο: πνευμονικές φλέβες (και επίσης το κοιλιακό τοίχωμα) - κίτρινο: το μεσοθωράκιο - βιολετί: το διάφραγμα



Κατάτμηση Εικόνας - Εφαρμογές

https://en.wikipedia.org/wiki/Image_segmentation

- Ανάκτηση εικόνας βάσει περιεχομένου
- Μηχανική όραση
- Ιατρική απεικόνιση (π.χ. τμηματοποίηση όγκου και άλλων παθολογιών, Ενδοχειρουργική πλοήγηση)
- Ανίχνευση αντικειμένων (π.χ. Ανίχνευση πεζών, Ανίχνευση προσώπου)
- Αναγνωριστικά καθήκοντα (Αναγνώριση προσώπου, Αναγνώριση δακτυλικών αποτυπωμάτων και ίριδας)
- Συστήματα ελέγχου της κυκλοφορίας
- Παρακολούθηση βίντεο

Κατάτμηση Εικόνας - Image Segmentation

- Αλγόριθμοι τμηματοποίησης για μονόχρωμες εικόνες γενικά βασίζονται σε μία από τις δύο βασικές ιδιότητες των τιμών έντασης εικόνας: ασυνέχεια και ομοιότητα.
- Στην πρώτη κατηγορία, η προσέγγιση είναι να χωρίσουμε μια εικόνα με βάση απότομες αλλαγές στην ένταση, όπως οι ακμές.
- Οι κύριες προσεγγίσεις στη $2^{η}$ κατηγορία βασίζονται στη διαίρεση μιας εικόνας σε περιοχές που είναι παρόμοιες σύμφωνα με μια σειρά προκαθορισμένων κριτηρίων.

Κατάτμηση Εικόνας με Κατωφλίωση

- Το κατώφλι ***T*** χωρίζει το προσκήνιο (με ένταση a) από το παρασκήνιο της εικόνας (με ένταση b) με την παρακάτω σχέση:

- $$g(x,y) = \begin{cases} a & \alpha\nu\ f(x,y) > T \\ b & \alpha\nu\ f(x,y) \leq T \end{cases}$$

- Το αποτέλεσμα όπως φαίνεται στο διπλανό παράδειγμα είναι το δέντρο να γίνεται τελείως μαύρο ενώ το χιόνι τελείως λευκό (=256)

https://en.wikipedia.org/wiki/Thresholding_(image_processing)

Μέθοδοι τμηματοποίησης κατωφλίωσης

1. Οι μέθοδοι με βάση το ιστόγραμμα, όπου, για παράδειγμα αναλύονται οι κορυφές, οι κοιλάδες και οι καμπυλώσεις του εξομαλυμένου ιστογράμματος για να βρεθεί ένα κατώφλι που να ξεχωρίζει διαφορετικές περιοχές.
2. Οι μέθοδοι που βασίζονται στη συσταδοποίηση, όπου με βάση τη φωτεινότητα των εικονοστοιχείων η εικόνα διαχωρίζεται σε ένα σύνολο από ομάδες με κοινά χαρακτηριστικά (π.χ. με τη μέθοδο k-means).
3. Οι μέθοδοι που βασίζονται σε χαρακτηριστικά αντικειμένων όπου ένα μέτρο καθορίζει την ομοιότητα μεταξύ π.χ. αρχικών (gray-level) και των δυαδικών (binarised) εικόνων, όπως η ομοιότητα των ασαφών σχημάτων, η σύγκριση των ακμών κλπ.
4. Οι χωρικές μέθοδοι που χρησιμοποιούν κατανομή πιθανότητας υψηλότερης τάξης και / ή συσχετισμό μεταξύ εικονοστοιχείων.
5. Οι τοπικές μέθοδοι προσαρμόζουν την τιμή κατωφλίου σε κάθε εικονοστοιχείο ανάλογα με τα χαρακτηριστικά τοπικής εικόνας. Σε αυτές τις μεθόδους, επιλέγεται διαφορετικό T για κάθε εικονοστοιχείο στην εικόνα εξετάζοντας κάθε φορά μια περιοχή γύρω από αυτό.

https://en.wikipedia.org/wiki/Thresholding_(image_processing)

Αλγόριθμος Τμηματοποίησης με Κατωφλίωση

1. Επιλέξτε μια αρχική εκτίμηση για το συνολικό όριο, T π.χ. τη μέση τιμή της εικόνας.
2. Τμηματοποιήστε την εικόνα χρησιμοποιώντας το T. Αυτό θα δημιουργήσει δύο ομάδες εικονοστοιχείων: το G1 είναι το σύνολο όλων των εικονοστοιχείων με τιμές έντασης μεγαλύτερες από T και G2 με τιμές μικρότερες ή ίσες με το T.
3. Υπολογίστε τις μέσες τιμές έντασης m_1 και m_2 για τα εικονοστοιχεία στις περιοχές G1 και G2, αντίστοιχα.
4. Υπολογίστε μια νέα τιμή κατωφλίου: $\mathrm{T} = \frac{1}{2}(m_1 + m_2)$

Αλγόριθμος Τμηματοποίησης με Κατωφλίωση

5. Επαναλάβετε τα βήματα 2 έως 4 μέχρι να εμφανιστεί η διαφορά Τ σε διαδοχικές επαναλήψεις μικρότερη από μια προκαθορισμένη τιμή, ΔΤ.
6. Τμηματοποιήστε την εικόνα χρησιμοποιώντας τη λειτουργία im2bw:

   g = im2bw (f, T/den)

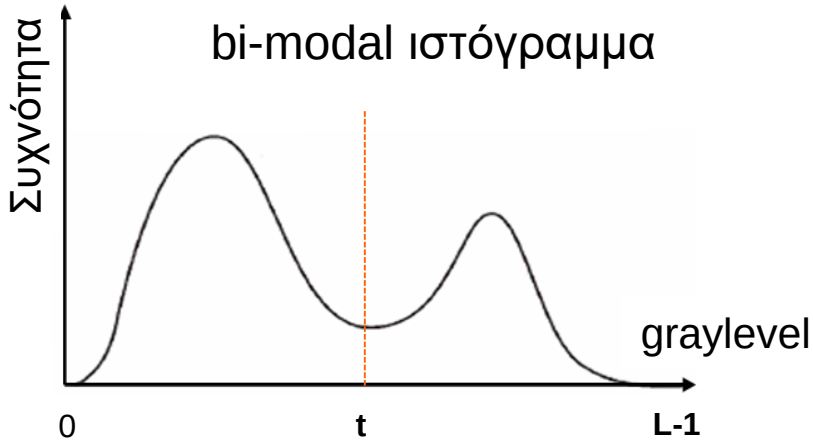
   Όπου den είναι ένας ακέραιος αριθμός (π.χ., 255 για μια εικόνα 8-bit) που κανονικοποιεί το μέγιστο του λόγου T/den στο 1, όπως απαιτείται από τη συνάρτηση im2bw.

Σχεδιάστε τον αλγόριθμο και δοκιμάστε τον στην εικόνα fingerprint κλπ.

Αλγόριθμος OtsuΤμηματοποίησης με Κατωφλίωση

- Ένα πολύ γνωστό παράδειγμα αυτόματης κατωφλίωσης είναι η μέθοδος του Otsu η οποία χρησιμοποιείται για την αυτόματη πραγματοποίηση ομαδοποίηση με βάση ένα κατώφλι.
- Ως αποτέλεσμα μια εικόνας graylevel μετατρέπεται σε δυαδική εικόνα.
- Ο αλγόριθμος υποθέτει ότι η εικόνα περιέχει δύο κατηγορίες εικονοστοιχείων που ακολουθούν το bi-modal ιστόγραμμα (pixel προσκηνίου και φόντου)

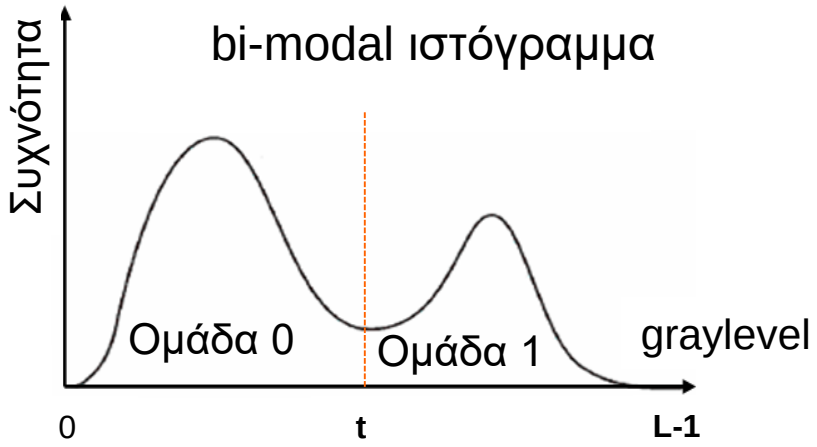
Στη συνέχεια υπολογίζει το βέλτιστο όριο διαχωρισμού των δύο κλάσεων έτσι ώστε η διακύμανση σε κάθε ομάδα να είναι ελάχιστη και η διακύμανση ανάμεσα στις δύο κλάσεις εικονοστοιχείων να είναι μέγιστη.



https://en.wikipedia.org/wiki/Otsu%27s_method

Αλγόριθμος Otsu Τμηματοποίησης με Κατωφλίωση

Στη συνέχεια υπολογίζει το βέλτιστο όριο διαχωρισμού των δύο κλάσεων έτσι ώστε η διακύμανση σε κάθε ομάδα να είναι ελάχιστη και η διακύμανση ανάμεσα στις δύο κλάσεις εικονοστοιχείων να είναι μέγιστη.



Η διακύμανση μεσα στις ομάδες δίνεται απο τη σχέση:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

https://en.wikipedia.org/wiki/Otsu%27s_method

- Τα σ_0^2, σ_1^2 είναι οι διακυμάνσεις των ομάδων 0,1 ενώ τα βάρη ω_0 $και$ ω_1 είναι οι πιθανότητες των δύο κλάσεων στην εικόνα που διαχωρίζονται με το κατώφλι t:

$$\omega_0 = \sum_{i=0}^{t-1} p_i \quad και\ \omega_1 = \sum_{i=t}^{\mathrm{L}-1} p_i$$

$όπου$ L είναι τα bins του ιστογράμματος (256 για 8bit εικόνες).

Αλγόριθμος Otsu

- *Ο Otsu δείχνει ότι η ελαχιστοποίηση της διακύμανσης μέσα στις ομάδες είναι το ίδιο με τη μεγιστοποίηση της διακύμανσης ανάμεσα στις δύο ομάδες :*

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_0(\mu_0 - \mu_\mathrm{T})^2 + \omega_1(\mu_1 - \mu_\mathrm{T})^2$$

- Όπου εξαρτάται από τις πιθανότητες ω και μέσους όρους μ κάθε κλάσης:

$$\mu_0 = \sum_{i=0}^{t-1} i\frac{p_i}{\omega_0} \quad \kappa\alpha\iota\ \mu_1 \sum_{i=t}^{\mathrm{L}-1} i\frac{p_i}{\omega_1} \quad \kappa\alpha\iota \quad \mu_\mathrm{T} = \sum_{i=0}^{\mathrm{L}-1} ip_i$$

https://en.wikipedia.org/wiki/Otsu%27s_method

Αλγόριθμος Otsu

- Υπολογίστε το ιστόγραμμα και τις πιθανότητες κάθε επιπέδου έντασης της εικόνας.
- Ορίστε τα αρχικά ω_i (0) και μ_i(0)
- Υπολογίστε όλα τα πιθανά κατώφλια από t=1… μέγιστη ένταση της εικόνας:
  - Ενημερώστε αλλαγές στα ω_i και μ_i
  - *Υπολογίστε την διακύμανση ανάμεσα στις δύο κλάσεις*
  - $\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$
- Το επιθυμητό όριο αντιστοιχεί στο μέγιστο $\sigma_b^2(t)$

$$\mu_0 = \sum_{i=0}^{t-1} i\frac{p_i}{\omega_0} \quad και\ \mu_1 \sum_{i=t}^{L-1} i\frac{p_i}{\omega_1} \quad και\ \mu_T = \sum_{i=0}^{L-1} ip_i$$

$$\omega_0 = \sum_{i=0}^{t-1} p_i \quad και\ \omega_1 = \sum_{i=t}^{L-1} p_i$$

https://en.wikipedia.org/wiki/Otsu%27s_method

Αλγόριθμος Otsu Matlab

```
function level = otsu(histogramCounts)
total = sum(histogramCounts); % '"total"' is the number of pixels in the given
image.
%% OTSU automatic thresholding method
sumB = 0;
wB = 0;
maximum = 0.0;
sum1 = dot( (0:255), histogramCounts);
for ii=1:256
    wB = wB + histogramCounts(ii);
    wF = total - wB;
    if (wB == 0 || wF == 0)
        continue;
    end
    sumB = sumB +  (ii-1) * histogramCounts(ii);
    mF = (sum1 - sumB) / wF;
    between = wB * wF * ((sumB / wB) - mF) * ((sumB / wB) - mF);
    if ( between >= maximum )
        level = ii;
        maximum = between;
    end
end
end
```

https://en.wikipedia.org/wiki/Otsu%27s_method

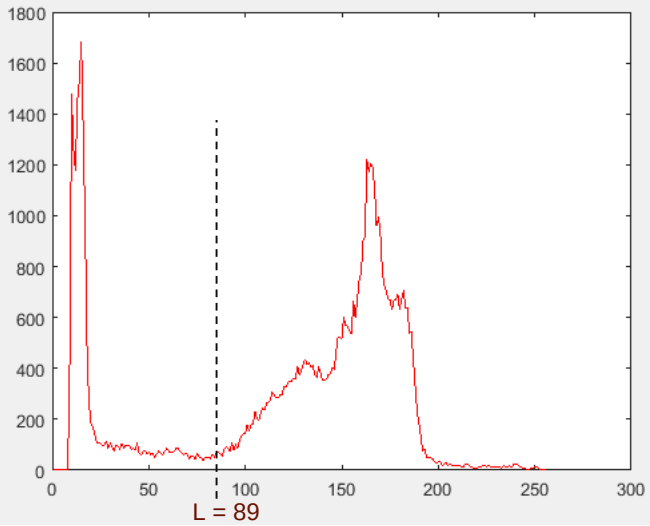
Αλγόριθμος Otsu Matlab

```
I=imread('cameraman.tif');
H=imhist(I);
L=otsu(H)

L =

    89

figure, imshow(I>L)
figure, plot(H,'r-')
figure, imshow(I)
```

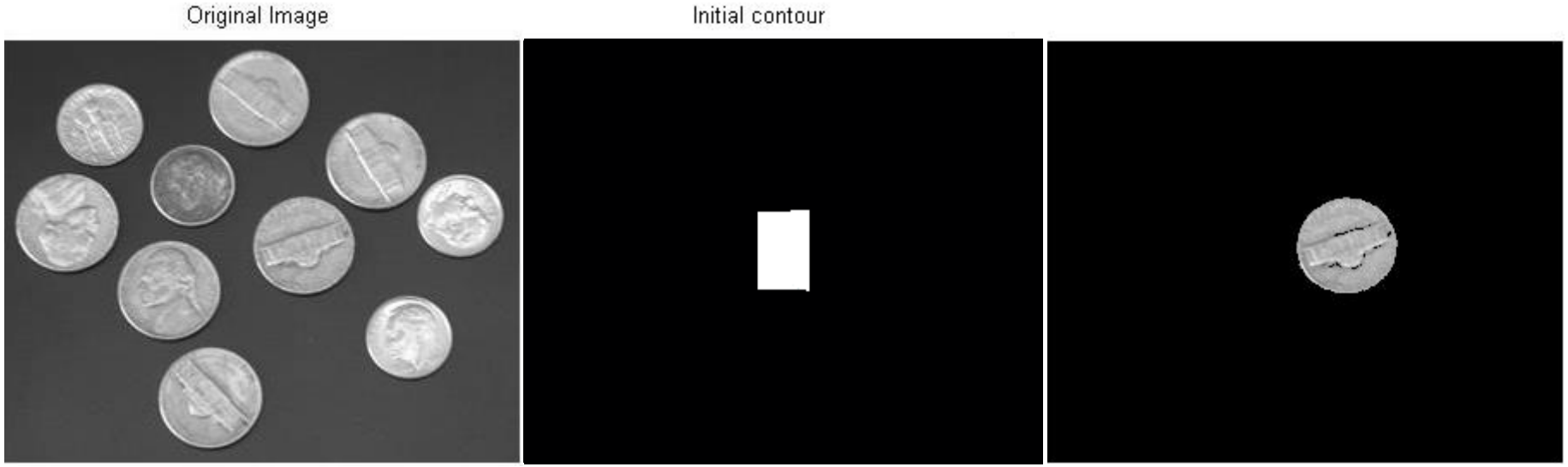


Άλλες μέθοδοι κατάτμησης

- Στις επόμενες διαφάνειες θα δούμε κάποια παραδείγματα κατάτμησης με τη μέθοδο active contours

Κατάτμηση Εικόνας με Matlab

Matlab




```
I = imread('coins.png');
imshow(I)
title('Original Image');
```

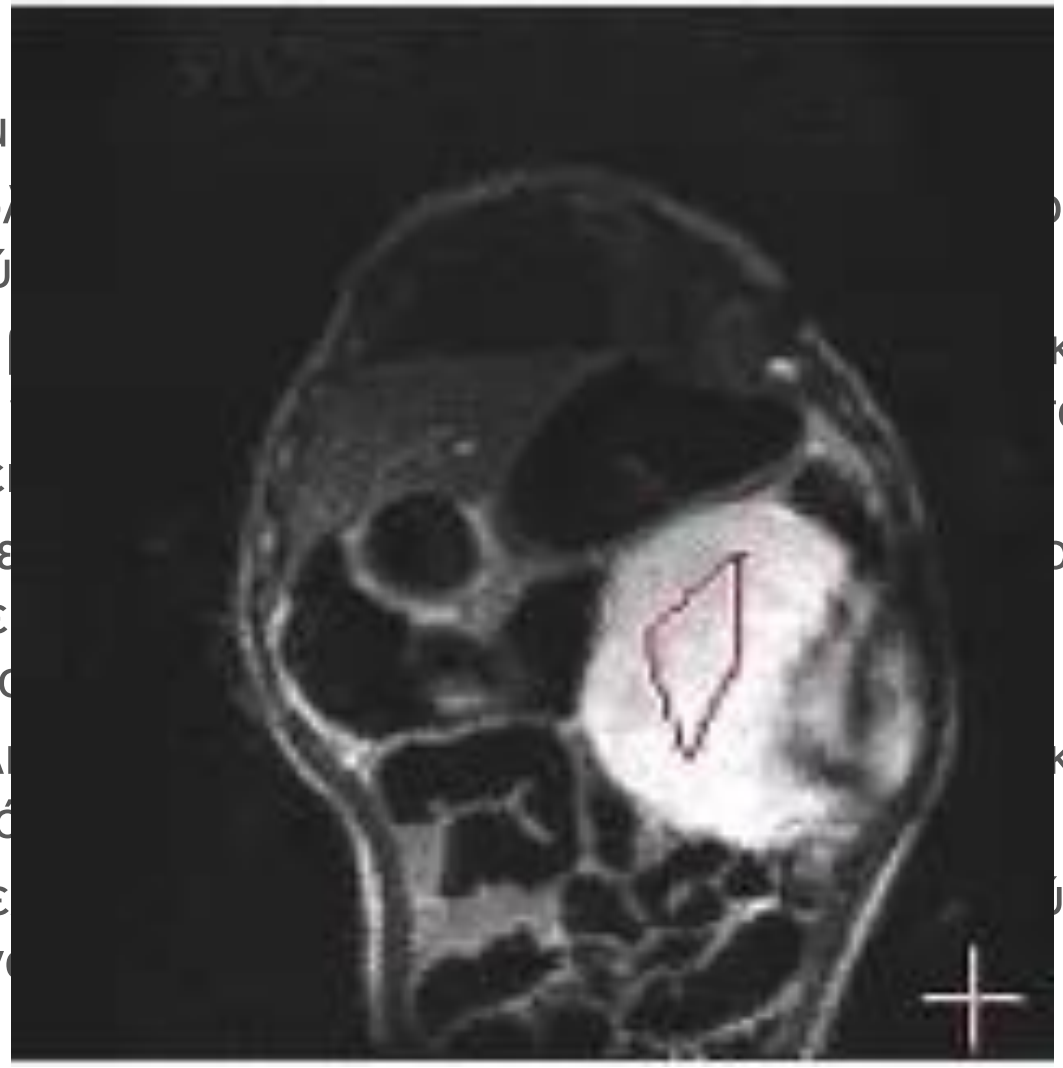
```
mask(100:end-100,135:end-135) = 1;
figure, imshow(uint8(mask).*I); title('Initial contour');
Iterations=zeros(size(I,1), size(I,2),100);
 for x=1:100
  Iterations(:,:,x)=uint8(activecontour(I,mask,x)).*I;
 end
X = permute(Iterations,[1 2 4 3]);
movie = immovie(X,map);
implay(movie);
```

Κατάτμηση καρκινικών περιοχών με active contours

- Η κατάτμηση των καρκινικών περιοχών με το χέρι είναι ανέφικτη σε μεγάλο αριθμό εικόνων και οδηγεί συχνά σε σφάλματα και σε συστηματικές διαφορές μεταξύ ειδικών.
- Ο σκοπός είναι να δημιουργήσουμε ένα εργαλείο που να βοηθάει το γιατρό στην κατάτμηση καρκίνου.
- Το εργαλείο πρέπει να εποπτεύεται από τον γιατρό ενώ την ίδια στιγμή να προσφέρει ένα ‘αντικειμενικό’ τρόπο τμηματοποίησης για περαιτέρω επεξεργασία (π.χ. εκτίμηση όγκου καρκίνου).

Προτεινόμενη Μέθοδος-Αποτελέσματα

Κατάτμηση καρκινικών περιοχών



- Τι είναι όμ
- Ένα σύνολ ουργούν μια καμπύ
- Η τεχνική καμπύλης με σκοπό του μας ενδιαφέρε
- Για να γίνε ου αντιστοιχε προσπαθ
- Η καμπύλ κά της εικόνας (ό
- Το ολικό ε νορο του αντικειμέν

Η μέθοδος Region Growing

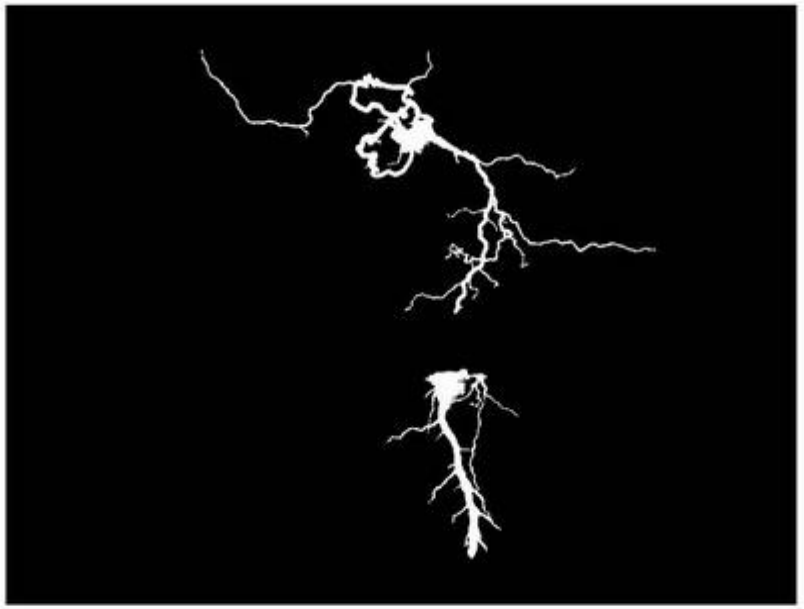
- Ξεκινώντας από ένα σημείο που δίνει ο χρήστης (seed point), η περιοχή αναπτύσσεται και μεγαλώνει διαδοχικά συγκρίνοντας όλα τα μη διατεθέντα γειτονικά εικονοστοιχεία στην περιοχή.
- Η διαφορά μεταξύ της τιμής έντασης ενός εικονοστοιχείου και του μέσου όρου της περιοχής γύρω από αυτό χρησιμοποιείται ως μέτρο ομοιότητας.
- Το εικονοστοιχείο με τη μικρότερη διαφορά που μετράται με αυτό τον τρόπο προστίθεται στην περιοχή.
- Αυτή η διαδικασία σταματά όταν η διαφορά έντασης μεταξύ μέσου όρου περιοχής και νέου εικονοστοιχείου γίνεται μεγαλύτερη από ένα συγκεκριμένο όριο (κατώφλι).

Η μέθοδος Region Growing

- Ο αλγόριθμος μπορεί να υλοποιηθεί και με πολλά seed points ου δίνει ο χρήστης και αναλόγως το κατώφλι της συνδεσιμότηταας με τη γειτονιά αλλάξει το αποτέλεσμα όπως φαίνεται στο παρακάτω παράδειγμα (από Wikipedia)

The original figure

The seed points: 255~255



Threshold: 190~255

https://en.wikipedia.org/wiki/Region_growing

Η μέθοδος Region Growing: Πλεονεκτήματα

1. Οι μέθοδοι ανάπτυξης περιοχής (region growing) μπορούν να διαχωρίσουν σωστά τις περιοχές που έχουν τις ίδιες ιδιότητες που ορίζουμε (gray level, textures, colors).
2. Μπορούν να πετύχουν στις πρωτότυπες εικόνες που έχουν σαφείς ακμές, καλά αποτελέσματα τμηματοποίησης.
3. Η ιδέα είναι απλή. Χρειαζόμαστε μόνο ένα μικρό αριθμό αρχικών σημείων για να αντιπροσωπεύσουμε την ιδιότητα στην οποία θέλουμε να βασιστούμε και στη συνέχεια να αναπτύξουμε την περιοχή.
4. Μπορούμε να προσδιορίσουμε με πολλαπλά κριτήρια τα seed points και τα κριτήρια που θέλουμε για την ανάπτυξη των περιοχών.

Η μέθοδος Region Growing Μειονεκτήματα

1. Υπολογιστικά δαπανηρή μέθοδος
2. Είναι μια τοπική μέθοδος χωρίς γενική άποψη του περιεχομένου της εικόνας και του προβλήματος που ασχολούμαστε.
3. Έχει ευαισθησία στο θόρυβο.
4. Εκτός αν η εικόνα έχει υποστεί πρώτα κατωφλίωση, μπορεί να υπάρχει μια συνεχής διαδρομή σημείων που σχετίζονται με το χρώμα που συνδέει οποιαδήποτε δύο σημεία της εικόνας.

Διαφορετικά seed points μπορεί να οδηγήσουν σε διαφορετικά αποτελέσματα κατωφλίωσης.

Η μέθοδος Region Growing- Υλοποίηση σε Matlab

```
% function to implement region growing
% BW=region(A,seedx,seedy)
% input parameters
% A image to implement the region growing
% seedx & seedy coordinates of the seed output parameters first rows then columns
% BW is the black&white image
```

```
function [BW] = myregiongrowing(A,seedx,seedy)

value=A(seedx,seedy);
amplitudT = 40;
[x y]=find(A<=(value + amplitudT/2)&A>=(value - amplitudT/2));
BW=zeros(size(A));
for i=1:length(x)
 BW(x(i),y(i))=1;
end

%figure, imagesc(BW), colormap gray
cc = bwconncomp(BW);
labeled = labelmatrix(cc);
regionLabel=labeled(seedx,seedy);
clear x y BW
BW=zeros(size(A));
[x y]=find(labeled==regionLabel);
for i=1:length(x)
 BW(x(i),y(i))=1;
end
```

Η μέθοδος Region Growing- Υλοποίηση σε Matlab

CC = bwconncomp(BW,CONN) specifies the desired connectivity for the connected components. CONN may have the following scalar values:

4	two-dimensional four-connected neighborhood
8	two-dimensional eight-connected neighborhood
6	three-dimensional six-connected neighborhood
18	three-dimensional 18-connected neighborhood
26	three-dimensional 26-connected neighborhood

Connectivity may be defined in a more general way for any dimension using a 3-by-3-by- ... -by-3 matrix of 0s and 1s. The 1-valued elements define neighborhood locations relative to the center element of CONN. CONN must be symmetric about its center element.

Η μέθοδος Region Growing- Υλοποίηση σε Matlab

L = labelmatrix(CC) creates a label matrix from the connected components structure CC returned by BWCONNCOMP.

The size of L is CC.ImageSize. The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object, the pixels labeled 2 make up a second object, and so on.

Η μέθοδος Region Growing- Υλοποίηση σε Matlab

```
A=imread('moon.tif');
% BW=myregiongrowing(A,335,69);
seedx=335;seedy=69;
value=A(seedx,seedy);
amplitudT = 40;
[x y]=find(A<=(value + amplitudT/2)&A>=(value - amplitudT/2));
BW=zeros(size(A));
for i=1:length(x)
 BW(x(i),y(i))=1;
end
figure, imshow(BW)
cc = bwconncomp(BW);
labeled = labelmatrix(cc);
figure, imshow(labeled)
figure, imshow(labeled,[])
regionLabel=labeled(seedx,seedy)
```

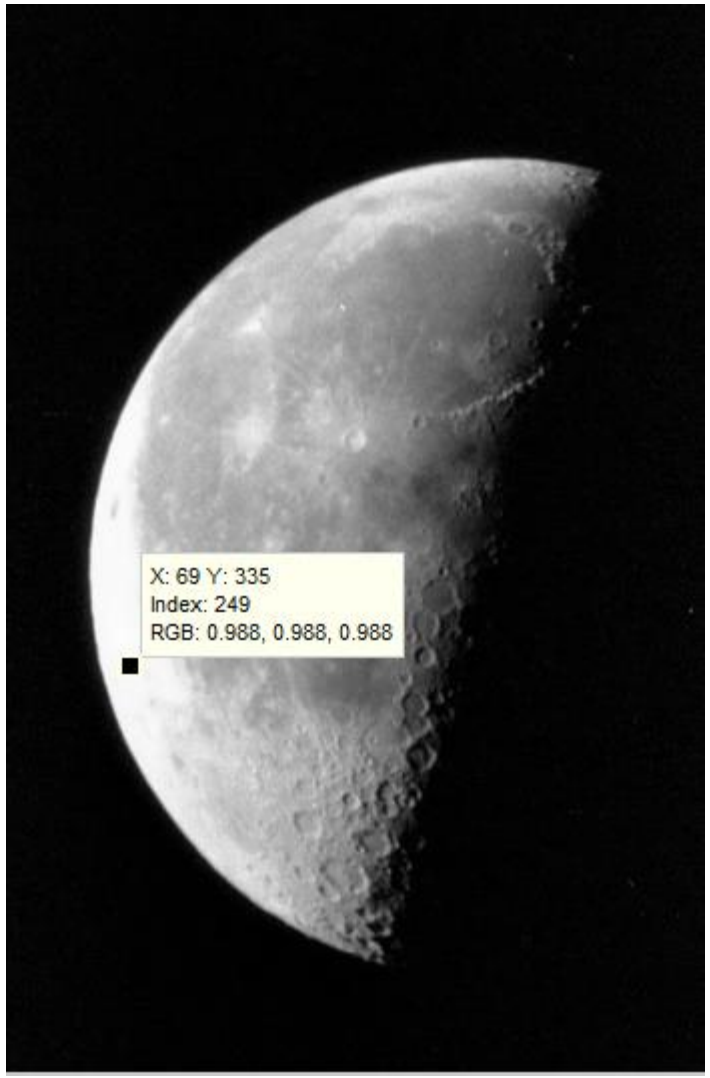
```
regionLabel =
```

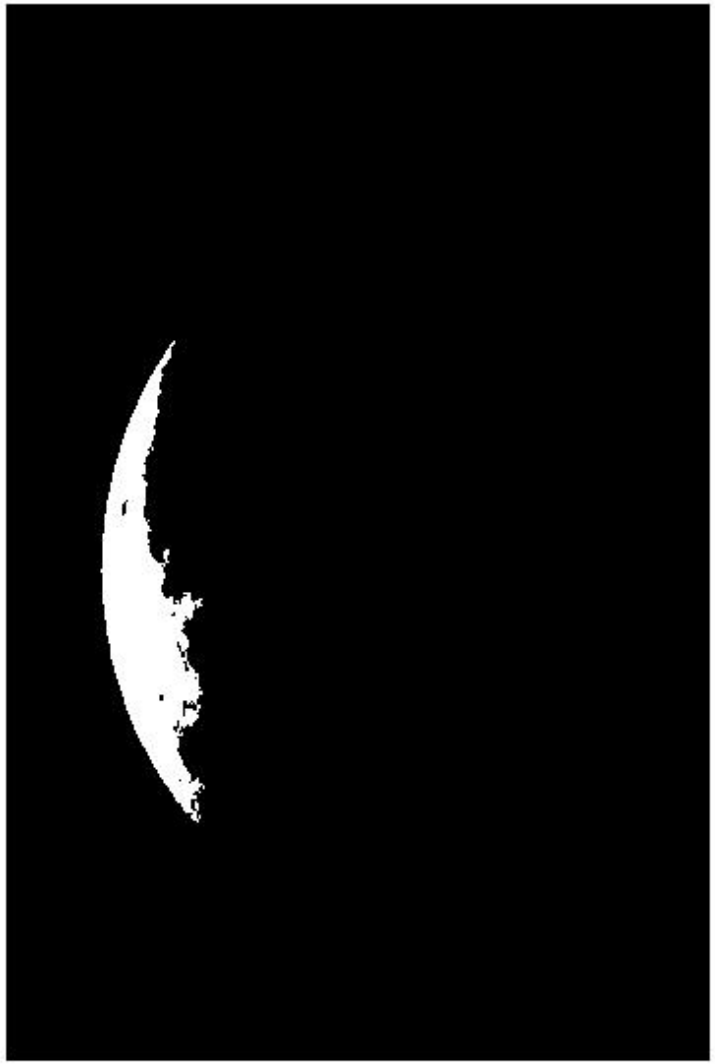
Επεξήγηση της συνάρτησης

Η μέθοδος Region Growing- Υλοποίηση σε Matlab

- A=imread('moon.tif');
- BW=myregiongrowing(A,335,69);
- figure, imshow(BW)

Η μέθοδος Region Growing- Υλοποίηση σε Matlab





Εφαρμογή kmeans σε Εικόνα

- Η χρήση του kmeans στην εικόνα μπορεί να θεωρηθεί ως μια μέθοδο τμηματοποίησης. Για να εφαρμόσουμε την τεχνική αυτή συσταδοποίησης σε μια εικόνα θα πρέπει να σκεφτούμε τι ακριβώς θέλουμε να κάνουμε:
- Αν έχουμε μονόχρωμη εικόνα (gray-scale) μπορούμε να θεωρήσουμε κάθε pixel ως ένα πρότυπο και να χωρίσουμε τα pixels της εικόνας σε όσες κλάσεις επιθυμούμε.
- Σε έγχρωμες εικόνες μπορούμε να χρησιμοποιήσουμε 2ή 3 κανάλια, δηλαδη 2 ή 3 χαρακτηριστικά ανα pixel για να κάνουμε kmeans.
- Για να το υλοποιήσουμε στη matlab σε κάθε περίπτωση πρέπει να κατανοήσουμε την εντολή reshape.

Εφαρμογή kmeans σε Εικόνα

- Ας πούμε ότι φορτώνουμε μια RGB εικόνα (ουσιαστικά είναι 3 πίνακες με τιμές που αντιπροσωπεύουν κάθε βασικό χρώμα).

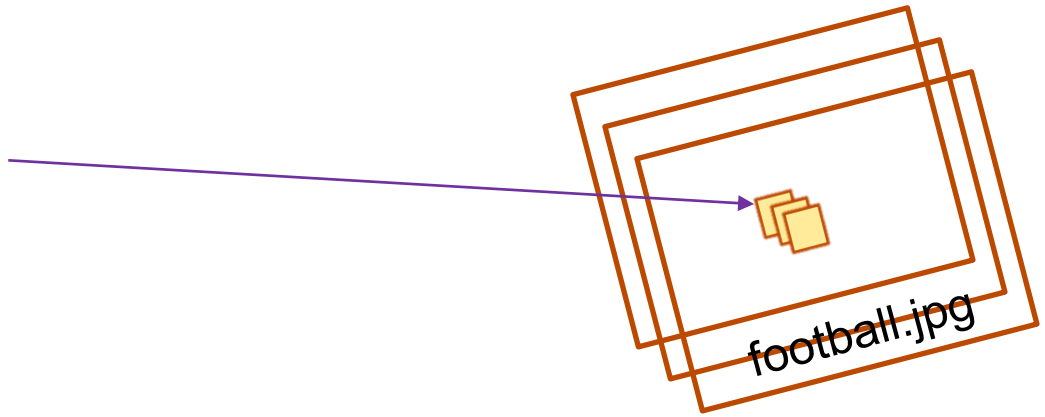
im = imread('football.jpg');

Στο workspace βλέπουμε ότι είναι <256x320x3 uint8>

Αν κόψουμε ένα μικρό 3Δ κύβο π.χ.:

test=im(1:4,1:4,1:3)

Τότε θα πάρουμε:



Εφαρμογή kmeans σε Εικόνα

test(:,:,1) =

49	43	33	44
36	38	39	38
34	30	36	37
37	24	27	42

test(:,:,2) =

86	80	73	84
73	75	79	78
71	67	73	74
74	61	64	79

test(:,:,3) =

113	107	99	110
100	102	105	104
98	94	100	101
101	88	91	106

Ας υποθέσουμε ότι θέλουμε το και τα 3 κανάλια πληροφορίας για να το περάσουμε για kmeans. Τότε γράφουμε:

ab = reshape(test,size(test,1)*size(test,2),3)

ab =

49	86	113
36	73	100
34	71	98
37	74	101
43	80	107
38	75	102
30	67	94
24	61	88
33	73	99
39	79	105
36	73	100
27	64	91
44	84	110
38	78	104
37	74	101
42	79	106

Διάνυσμα με 3 χαρακτηριστικά που αντιστοιχούν στο pixel (1,1)

Εφαρμογή kmeans σε Εικόνα

test(:,:,1) =

49	43	33	44
36	38	39	38
34	30	36	37
37	24	27	42

test(:,:,2) =

86	80	73	84
73	75	79	78
71	67	73	74
74	61	64	79

test(:,:,3) =

113	107	99	110
100	102	105	104
98	94	100	101
101	88	91	106

Ας υποθέσουμε ότι θέλουμε το 2 και 3 κανάλι πληροφορίας για να το περάσουμε για kmeans. Τότε γράφουμε:
test2=test(:,:,2:3)

test2(:,:,1) =

86	80	73	84
73	75	79	78
71	67	73	74
74	61	64	79

test2(:,:,2) =

113	107	99	110
100	102	105	104
98	94	100	101
101	88	91	106

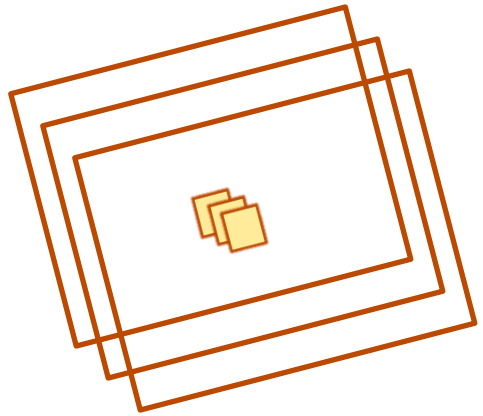
Για να μπορούμε όμως να τα περάσουμε στον αλγόριθμο kmeans θα πρέπει να χρησιμοποιήσουμε την εντολή reshape ώστε κάθε pixel να έχει 2 χαρακτηριστικά. Τότε γράφουμε:

ab = reshape(test2,size(test2,1)*size(test2,2),2)

ab =

86	113
73	100
71	98
74	101
80	107
75	102
67	94
61	88
73	99
79	105
73	100
64	91
84	110
78	104
74	101
79	106

→ Διάνυσμα με 2 χαρακτηριστικά που αντιστοιχούν στο pixel (1,1)



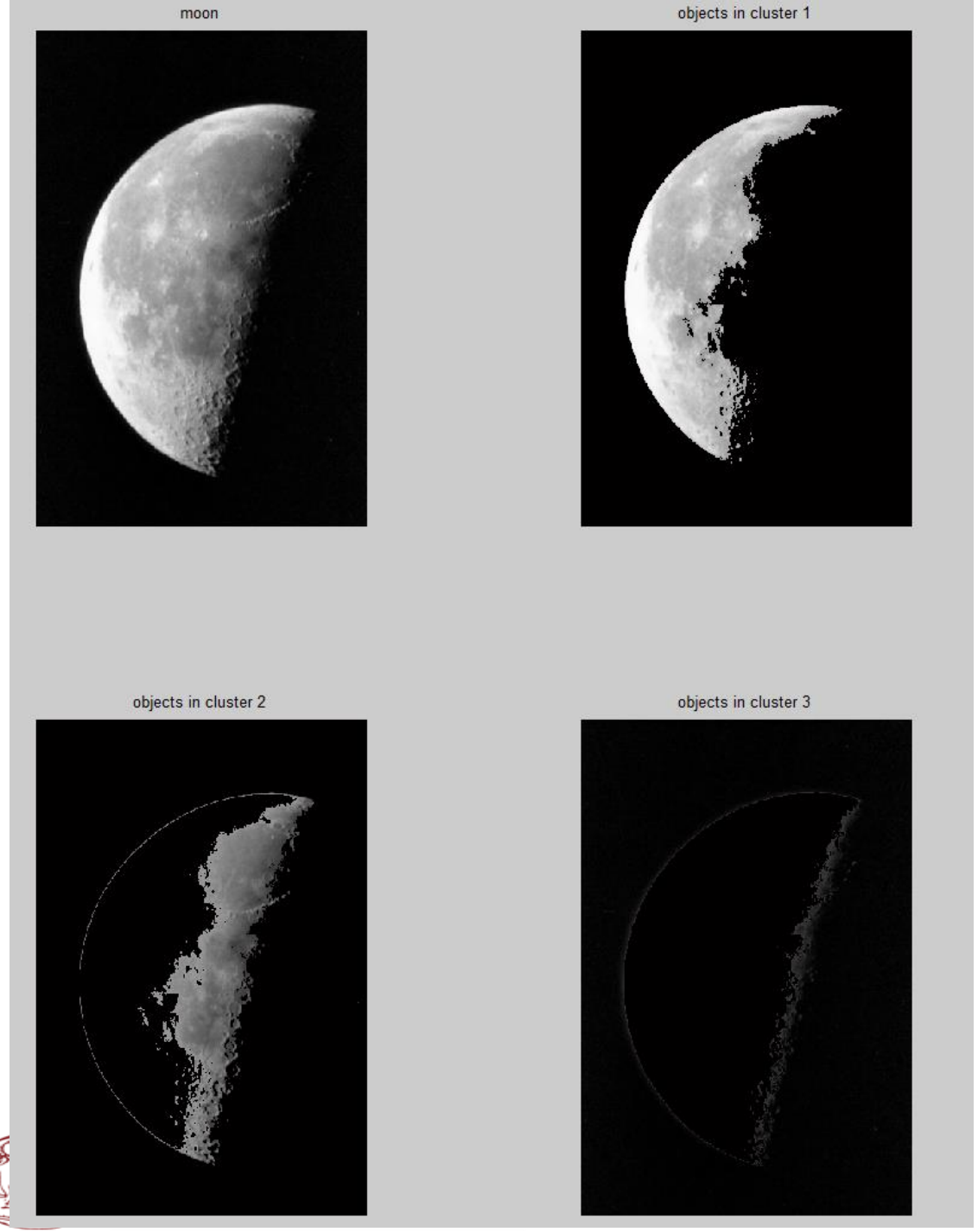
Εφαρμογή kmeans σε Εικόνα

- Ας ξεκινήσουμε από μια ασπρόμαυρη εικόνα του φεγγαριού. Με τον τρόπο που περιγράψαμε μπορούμε να περάσουμε στο kmeans την ένταση κάθε pixel ως το χαρακτηριστικό του προτύπου.
- Τα αποτελέσματα και ο κώδικας που φτιάξαμε φαίνονται στην επόμενη διαφάνεια όπου χρησιμοποιούμε τον αλγόριθμο kmeans για να τμηματοποιήσουμε την εικόνα σε K=3 κλάσεις.

Άσκηση Εφαρμογή kmeans σε Εικόνα

- Α.Να εφαρμόσετε τη μέθοδο kmeans για να τμηματοποιήσετε την gray scale ασπρόμαυρη εικόνα moon.tif σε τρείς κάλσεις.
- Β.Να φορτώσετε την RGB εικόνα της Matlab football.jpg και να εφαρμόσετε την μέθοδο kmeans χρησιμοποιώντας α) Όλα τα κανάλια και β)Μόνο τα κανάλια G,B. Να γίνει τμηματοποίηση για K=3 και K=4 σε όλα τα παραπάνω και να γραφτεί αναφορά με κώδικα, εξηγήσεις, αποτελέσματα εικόνων και γενικά σχόλια.

Οι απαντήσεις είναι στα επόμενα slides (μόνο για διδάσκοντες ☺)




```
im=imread('moon.tif');


nrows = size(im,1);
ncols = size(im,2);
ab = reshape(double(im),nrows*ncols,1);

nColors = 3;
% repeat the clustering 5 times to avoid local minima
[cluster_idx, cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean', ...
                                    'Replicates',5);


pixel_labels = reshape(cluster_idx,nrows,ncols);
figure, imshow(pixel_labels,[]), title('image labeled
by cluster index');


figure
subplot(2,2,1), imshow(im), title('moon');

subplot(2,2,2), imshow(im.*(uint8(pixel_labels==1))),
title('objects in cluster 1');
subplot(2,2,3), imshow(im.*(uint8(pixel_labels==2))),
title('objects in cluster 2');
subplot(2,2,4), imshow(im.*(uint8(pixel_labels==3))),
title('objects in cluster 3');
```

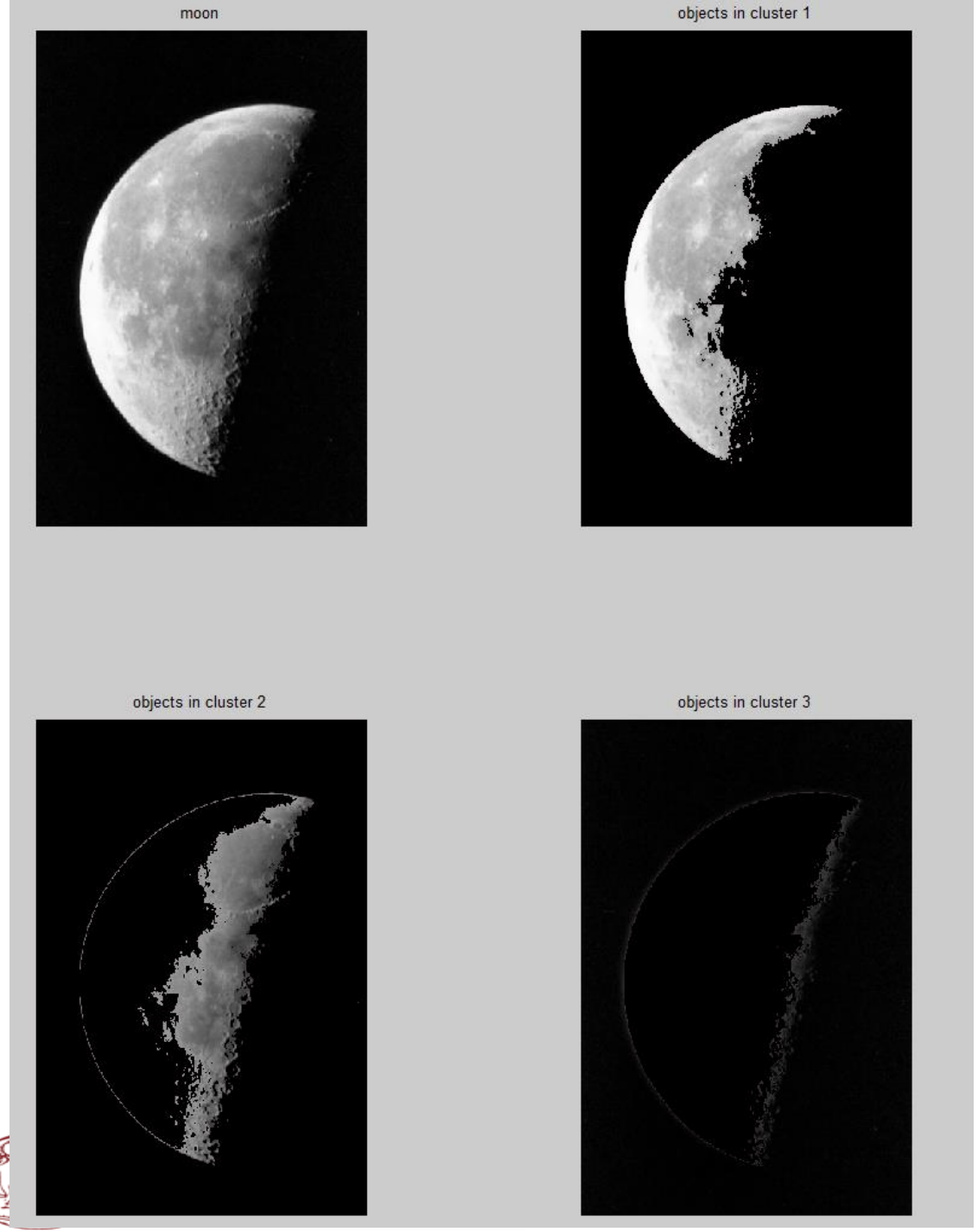
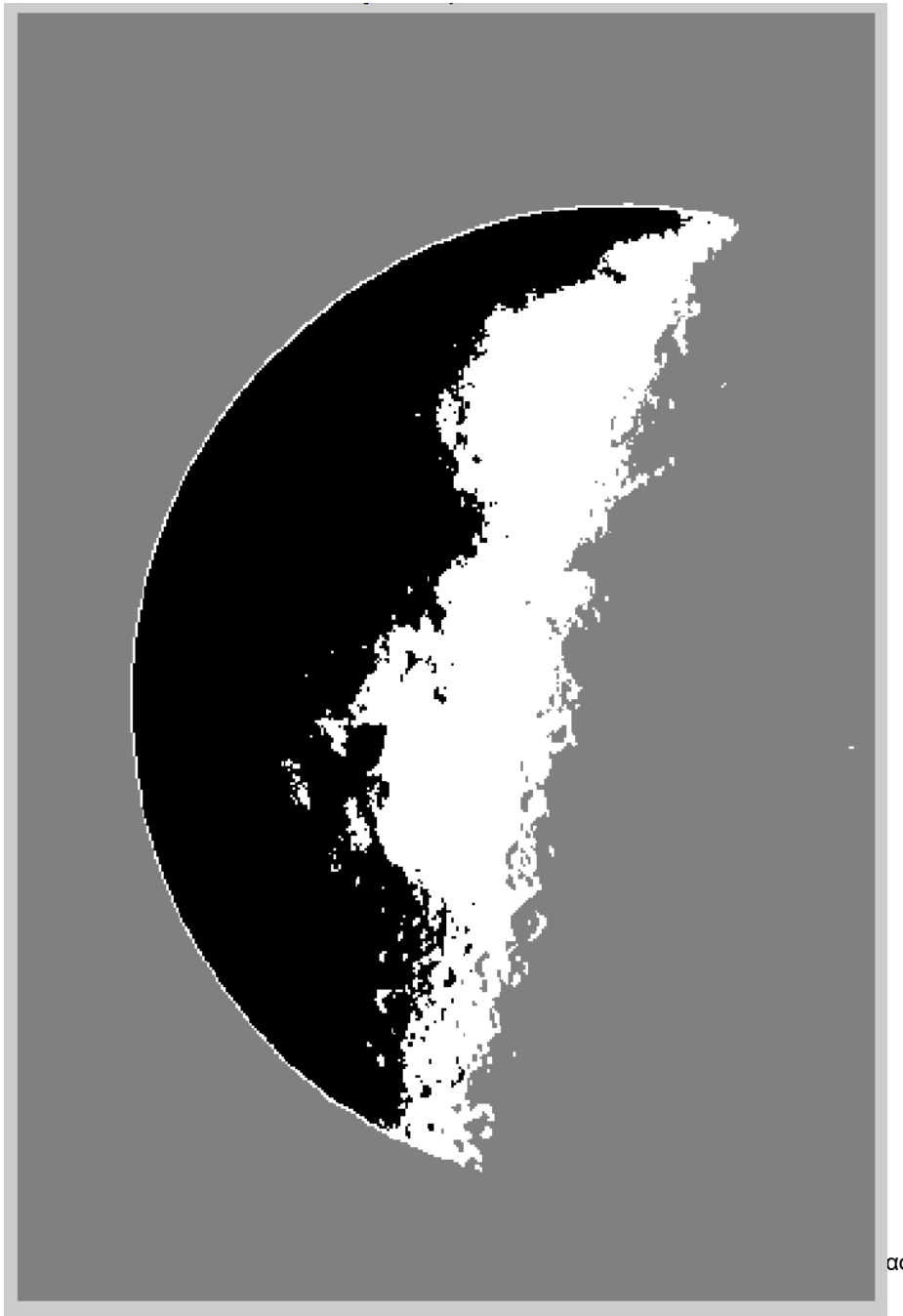
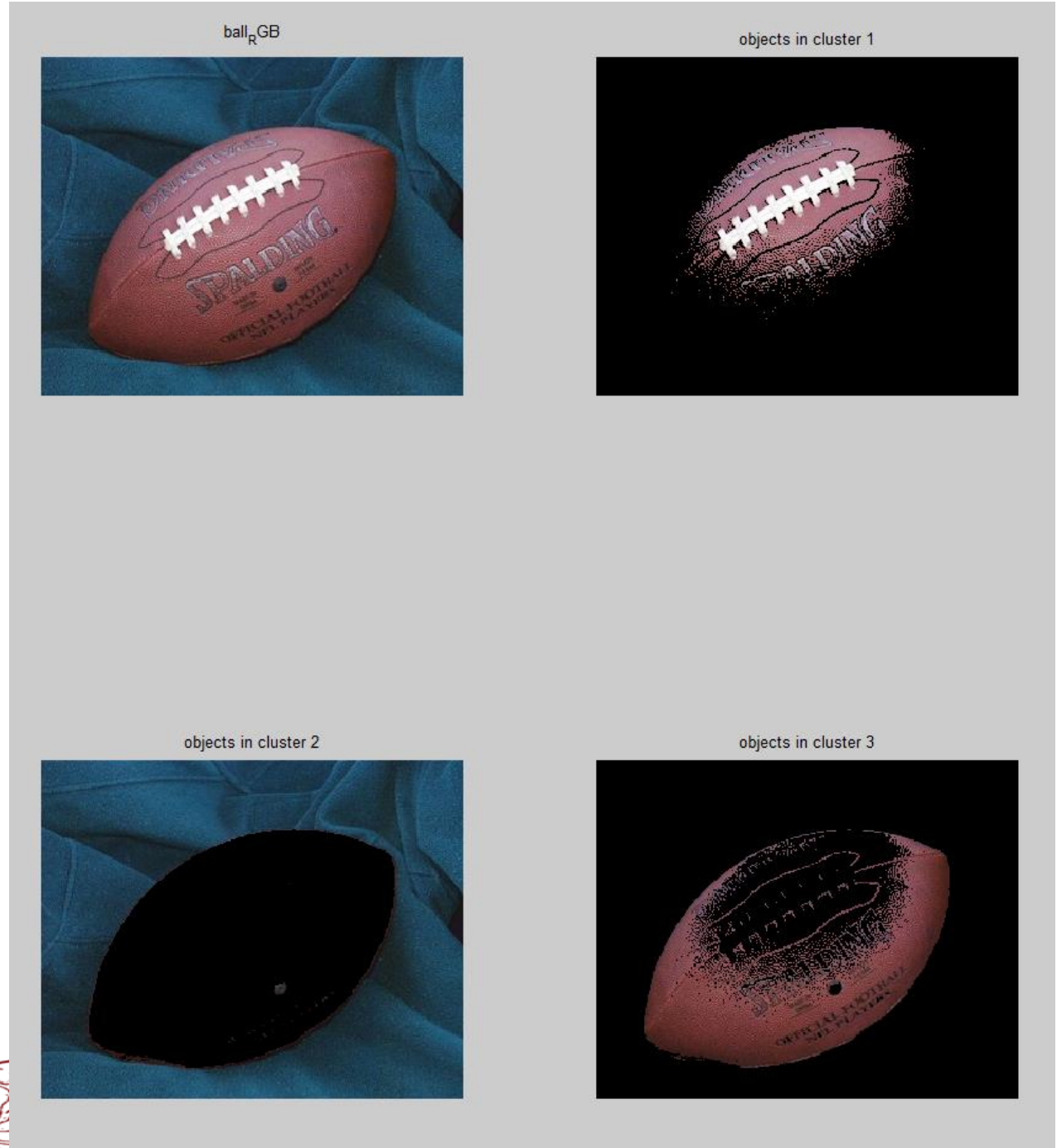
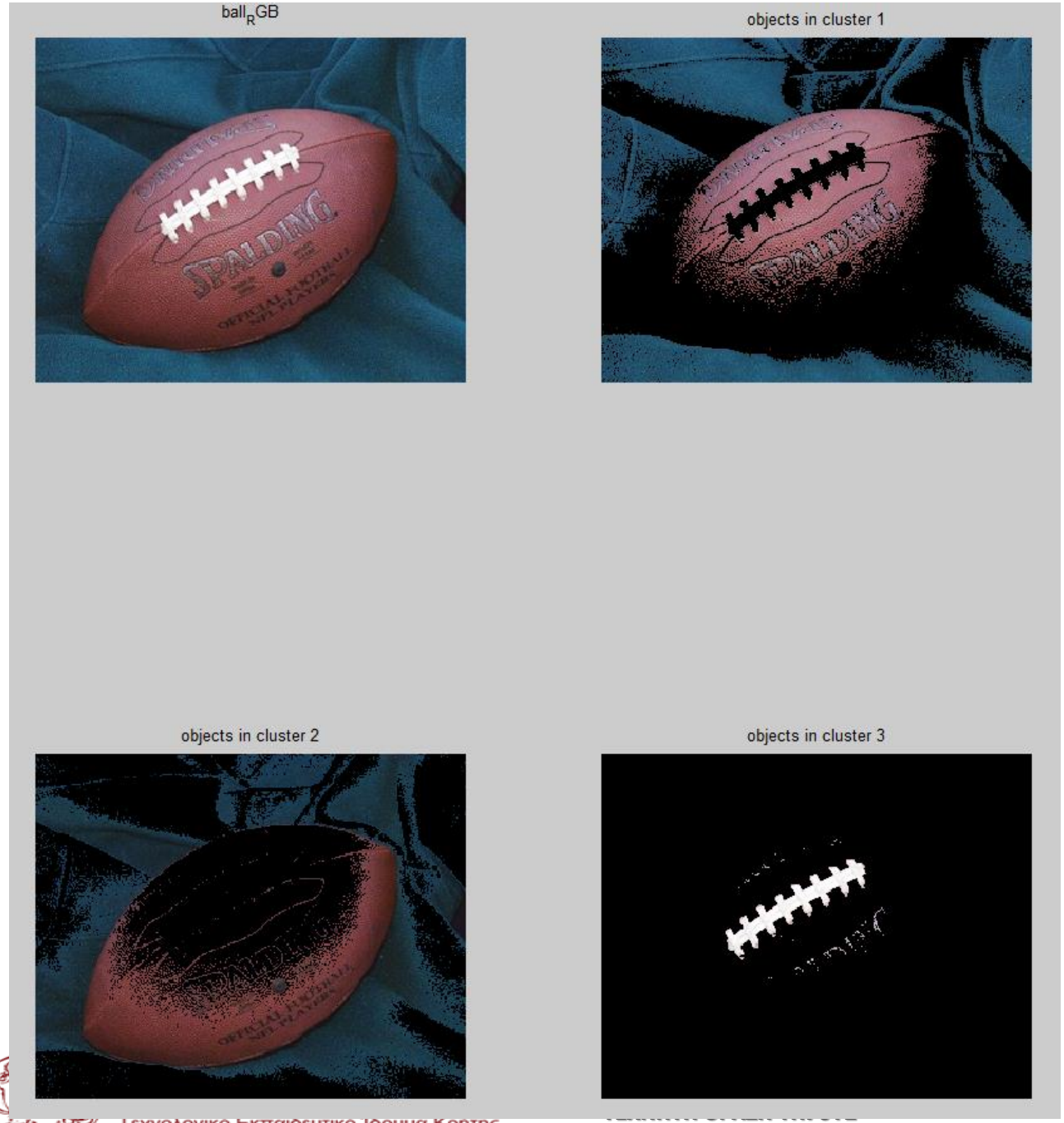


Image Labelled by Cluster Index






```
im = imread('football.jpg');
imshow(im), title('ball_RGB');
ab = double(im(:,:,1:3));
 nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,3);
 nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                       'Replicates',3);
 pixel_labels = reshape(cluster_idx,nrows,ncols);
figure, imshow(pixel_labels,[]), title('image labeled by cluster index');
segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);
for k = 1:nColors
    color = im;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end
figure
subplot(2,2,1), imshow(im), title('ball_RGB');
subplot(2,2,2), imshow(segmented_images{1}), title('objects in cluster 1');
subplot(2,2,3), imshow(segmented_images{2}), title('objects in cluster 2');
subplot(2,2,4), imshow(segmented_images{3}), title('objects in cluster 3');
```




```
im = imread('football.jpg');
imshow(im), title('ball_RGB');
ab = double(im(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
 nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                       'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
figure, imshow(pixel_labels,[]), title('image labeled by cluster index');
  segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);


for k = 1:nColors
    color = im;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end
figure
subplot(2,2,1), imshow(im), title('ball_RGB');
subplot(2,2,2), imshow(segmented_images{1}), title('objects in cluster 1');
subplot(2,2,3), imshow(segmented_images{2}), title('objects in cluster 2');
subplot(2,2,4), imshow(segmented_images{3}), title('objects in cluster 3');
```