

## **3η Εργαστηριακή Άσκηση**

---

**Εντοπισμός Χωρο-χρονικών Σημείων Ενδιαφέροντος και  
Εξαγωγή Χαρακτηριστικών σε Βίντεο Ανθρωπίνων  
Δράσεων**

**Μάθημα : Όραση Υπολογιστών**



**Ροή Σ**

Συνεργάτες :

- Βαβουλιώτης Γεώργιος ( Α.Μ. : 03112083 )
- Σταυρακάκης Δημήτριος ( Α.Μ. : 03112017 )

**Εισαγωγή-Σκοπός Άσκησης :** Στην τρίτη εργαστηριακή άσκηση αυτό με το οποίο θα ασχοληθούμε είναι η εξαγωγή χωρο-χρονικών χαρακτηριστικών με σκοπό την εφαρμογή τους στο πρόβλημα της κατηγοριοποίησης video με ανθρώπινες εκφράσεις. Όπως είδαμε και στην πρώτη εργαστηριακή άσκηση(της οποίας τα συμπεράσματα είναι πολύ χρήσιμα για την άσκηση αυτή) η συλλογή των τοπικών χαρακτηριστικών(local features) έχουν συμβάλλει σημαντικά στην αναγνώριση αντικειμένων αλλά επίσης κυριαρχεί και στο τομέα αναγνώρισης ανθρωπίνων δράσεων, γεγονός το οποίο αποτελεί και το επίκεντρο της άσκησης αυτής. Με την χρήση των local features καταφέρνουμε να μειώσουμε τη διάσταση κάθε βίντεο αλλά επίσης καταφέρνουμε να τα μετασχηματίσουμε με τέτοιο τρόπο ώστε να είναι πλέον κατηγοριοποιήσιμα. Συγκεκριμένα έχουμε στη διάθεση μας 3 κλάσεις ανθρωπίνων δράσεων (walking, running, boxing) για τις οποίες θα πρέπει να εξάγουμε χωρο-χρονικούς περιγραφητές με σκοπό να γίνει σωστή κατηγοριοποίηση.

## Μέρος 1 : Χωρό-χρονικά Σημεία Ενδιαφέροντος

Στο πρώτο μέρος της άσκησης καλούμαστε να υλοποιήσουμε δυο διαφορετικούς ανιχνευτές τοπικών χαρακτηριστικών, τον **Harris detector** και τον **Gabor detector**. Στη συνέχεια για καθένα από τους δυο αυτούς ανιχνευτές υπολογίζουμε τα σημεία ενδιαφέροντος σαν τα τοπικά μέγιστα του κριτηρίου σημαντικότητας. Θα επιστρέφουμε τα σημεία με τις μεγαλύτερες τιμές του κριτηρίου σημαντικότητας, το οποίο φαίνεται στη συνέχεια της αναφοράς, δηλαδή επιλεγούμε τοπικά μεγιστα που ξεπερνούν ενα threshold. Δηλαδή παίρνουμε όσα πληρουν τα κριτηρια(να ειναι τοπικα μεγιστα και να ξεπερνούν ενα threshold) ή αν αυτα υπερβαινουν τα 600 τότε παίρνουμε τα 600 μεγαλυτερα εξ αυτων .

Για την υλοποίηση του **Harris detector** ακολουθήσαμε την εξής πορεία με την βοήθεια του Matlab:

Για να την υλοποίηση του χρησιμοποιήσαμε την λογική με την οποια υλοποιήσαμε τον ανιχνευτή γωνιών Harris-Stephens στην 1η εργαστηριακή άσκηση, επεκτείνοντας τον απλά στις 3 διαστάσεις και κάνοντας τις απαραίτητες τροποποιήσεις. Από άποψη μαθηματικών αυτό που κάναμε είναι να υπολογίζουμε τον 3X3 πίνακα M(x,y,t) σύμφωνα με τον παρακάτω τύπο :

$$M(x, y, t; \sigma, \tau) = g(x, y, t; s\sigma, s\tau) * (\nabla L(x, y, t; \sigma, \tau)(\nabla L(x, y, t; \sigma, \tau))^T)$$

όπου :

- $g(x, y, t; s\sigma, s\tau)$  : 3D Gaussian πυρήνας ομαλοποιήσης.
- $\nabla L(x, y, t; \sigma, \tau)$  : χωρο-χρονικές παράγωγοι για το σ(χωρική κλίμακα) και το t(χρονική κλίμακα).

Ο πίνακας M αυτό που κάνει είναι να περιγράφει το *local gradient distribution*, χωρικά σε κλίμακα σ και χρονικά σε κλίμακα τ.

Για να καταφέρουμε να υπολογίσουμε εύκολα τις παραπάνω χωρο-χρονικές παραγώγους χρησιμοποίσαμε συνέλιξη με τον πυρήνα κεντρικών διαφορών προσαρμοσμένο κάθε φορά στην διάσταση που επιθυμούμε.

Τέλος το κριτήριο γωνιότητας που χρησιμοποιούμε είναι το εξής :

$$H(x, y, t) = \det(M(x, y, t)) - k \cdot \text{trace}^3(M(x, y, t))$$

Η συνάρτηση  $\mathbf{H(x,y,t)}$  ονομάζεται *saliency function* σύμφωνα με τον βιολογικό όρο saliency, ο οποίος περιγράφει το interestingness περιοχών της εικόνας. Στο Harris detector παίρνουμε μέγιστα 3D σφαίρας με ακτίνα 2 που ξεπερνούν το 0.005 της μέγιστης τιμής του  $\mathbf{H(x,y,t)}$  όπως φαίνεται και στον κώδικα της άσκησης. Παραπάνω λεπτομέριες για τον τρόπο υλοποίησης του **Harris detector** μπορείτε να δείτε στον source code για το μέρος 1 που σας επισυνάπτω στο zip αρχείο.

Θα μπορούσαμε να προσθέσουμε οτι στον **Harris detector**, ο άξονας του χρόνου δεν είναι απλά μια τρίτη διάσταση της εικόνας, αλλά περιγράφει μια πολύ διαφορετική οντότητα. Συγκεκριμένα διαπιστώθηκε ότι ο ανιχνευτής Harris μπορεί να οδηγήσει σε μη ικανοποιητικά αποτελέσματα, δεδομένου ότι τείνει να παράγει πολύ λίγα σημεία ενδιαφέροντος . Το γεγονός αυτό προκάλεσε την ανάπτυξη του περιοδικού ανιχνευτή τον οποίο αναλύω παρακάτω και είναι ο Gabor detector, τον οποίο επινόησε ο Dollar.

Για την υλοποίηση του **Gabor detector** ακολουθήσαμε την εξής πορεία με την βοήθεια του Matlab:

Για την υλοποίηση αυτού του detector αρχικά εξομαλύνουμε το video στις χωρικές διαστάσεις με την χρήση ενός 2D Gaussian πυρήνα με τυπική απόκλιση σ. Στη συνέχεια αυτό που κάνουμε είναι να εφαρμόσουμε ενα χρονικό φίλτραρισμα στο εξομαλυμένο video με χρήση ενός ζεύγους φίλτρων Gabor, των οποίων οι κρουστικές αποκρίσεις φαίνονται παρακάτω σε αναλυτική μορφή και στην επόμενη σελίδα η μορφή τους(η δεξιά είκονα) με την βοήθεια του Matlab:

$$h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega) \exp(-t^2/2\tau^2) \text{ και } h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega) \exp(-t^2/2\tau^2)$$

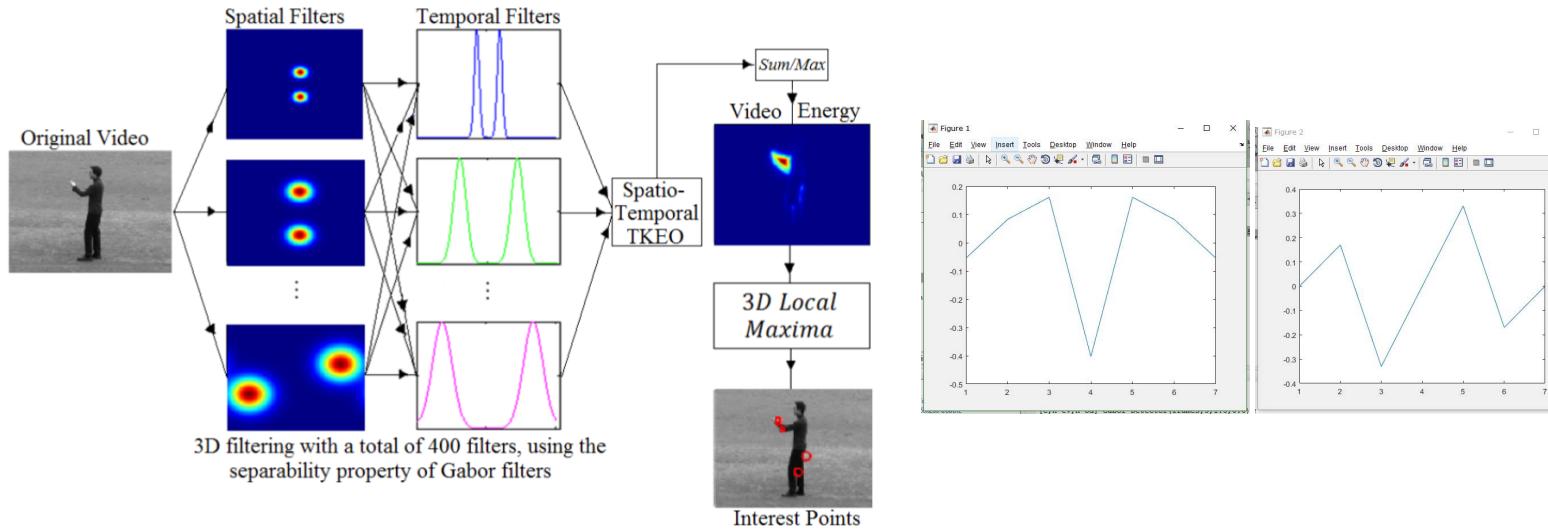
Για τον υπολογισμό αυτό θεωρήσαμε μέγεθος παραθύρου [-2τ,2τ] και κανονικοποιήσαμε με την νόρμα L1.

Τέλος το κριτήριο σημαντικότητας που χρησιμοποιούμε προκύπτει από την τετραγωνική ενέργεια της εξόδου για τα δυο Gabor φίλτρα, όπως φαίνεται και από τον παρακάτω μαθηματικό τύπο :

$$H(x, y, t) = (I(x, y, t) * g * h_{ev})^2 + (I(x, y, t) * g * h_{od})^2$$

Στον Gabor detector παίρνουμε μέγιστα 3d σφαίρας με ακτίνα 2 που ξεπερνούν το 0.05 της μέγιστης ενέργειας που βρίσκουμε (οι παράμετροι αυτοί που δίνουμε στις κλήσεις των συναρτήσεων).

Μια απεικόνιση αυτών που ανέφερα παραπάνω φαίνεται στην παρακάτω εικόνα(την αριστερή εικόνα) ποιοτικά, για να κατανοηθεί πλήρως η πορεία που ακολουθούμε(το video εισόδου είναι ενδεικτικά ενα boxing action video, όπως αυτό της που μας δίνεται) :



Παραπάνω λεπτομέριες για τον τρόπο υλοποίησης του **Gabor detector** μπορείτε να δείτε στον source code για το μέρος 1 που σας δίνω στο zip αρχείο.

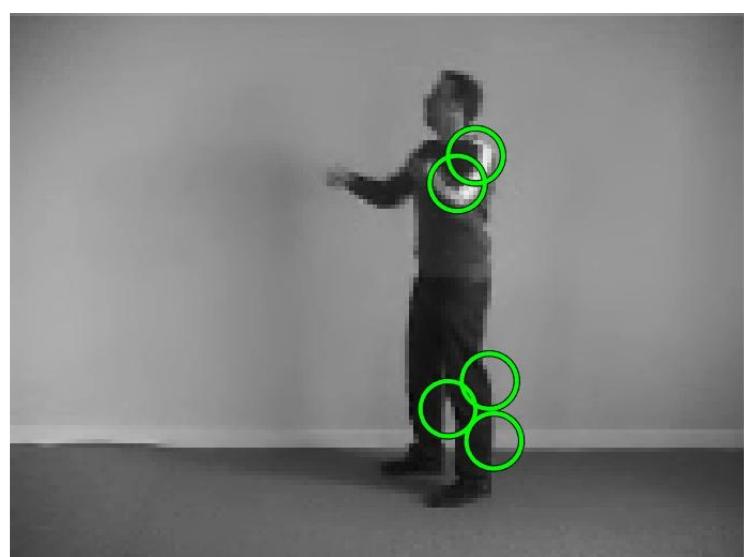
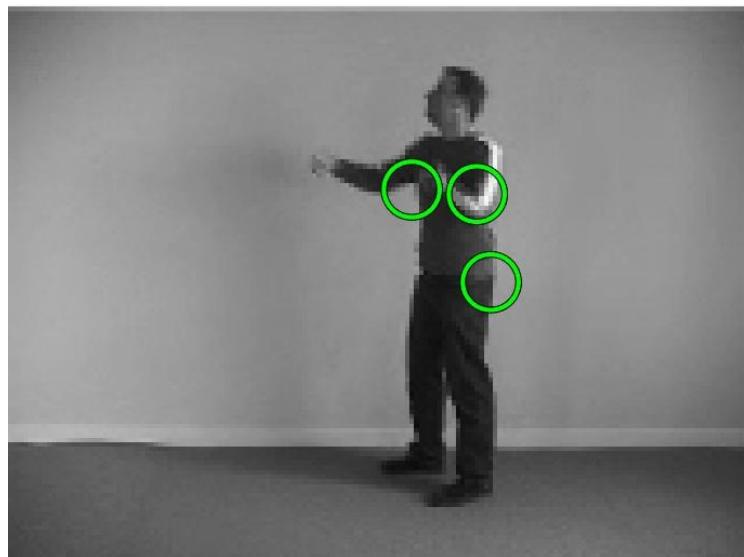
Ενημερωτικά και μόνο αναφέρω οτι υπάρχει και ο **Learned detector** ο οποίος σε αντίθεση με τους δυο παραπάνω ανιχνευτές σημείων ενδιαφέροντος, βασίζεται στα χαρακτηριστικά της εικόνας τα οποία επιλέγονται από το ανθρώπινο οπτικό σύστημα.

**Σημαντική Παρατήρηση :** Για την ευρεση τοπικου μεγιστου καναμε dilation με πυρηνα που δίνεται από το matlab με την εντολη: **B\_sq = strel('sphere', 2)**; εντολη που υποστηρίζεται πλεον στο Matlab 2016a οπου υλοποιήθηκε ο κώδικας. Σε παλαιότερη έκδοση για να τρέξει θα πρέπει να αντικατασταθεί η εν λόγω εντολή με τις εξής εντολές :  
**[x, y, z] = ndgrid(-2:2);**  
**B\_sq = strel(sqrt(x.^2 + y.^2 + z.^2) <= 2);**

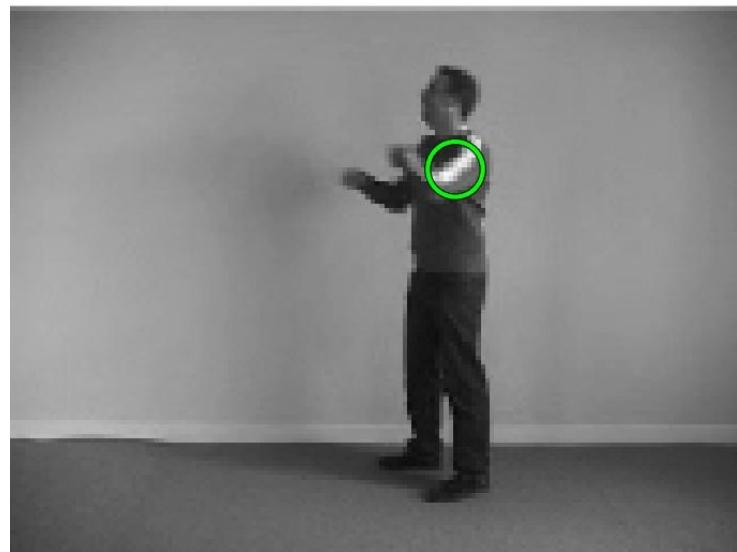
Αφού πλεόν έχουμε υλοποιήσει και τους δυο detectors μπορούμε να εξάγουμε τα σημεία ενδιαφέροντος ως τα τοπικά μέγιστα του κριτηρίου σημαντικότητας. Στον Harris χρησιμοποιήσαμε  $\sigma=2$  και  $\tau=0.7$  και στον Gabor  $\sigma=2$  και  $\tau=1.5$ . Ενδεικτικά σας παρουσιάζω στη συνέχεια για κάποια frames τα αποτελέσματα τα οποία πήραμε και με τους δυο detectors:

Για το **boxing** χρησιμοποίησα από video με όνομα **person06\_boxing\_d4\_uncomp** και συγκεκριμένα επέλεξα να σας δείξω τα frames 18,19,20(από πάνω προς τα κάτω) :

### Harris Detector Results :

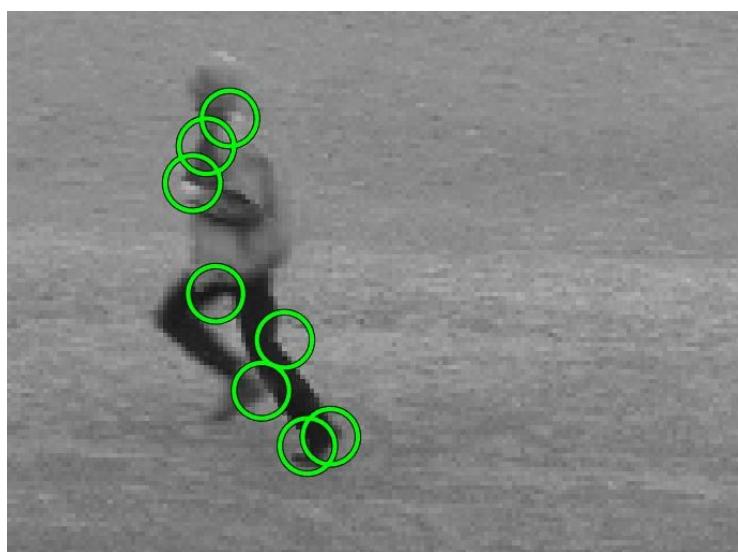
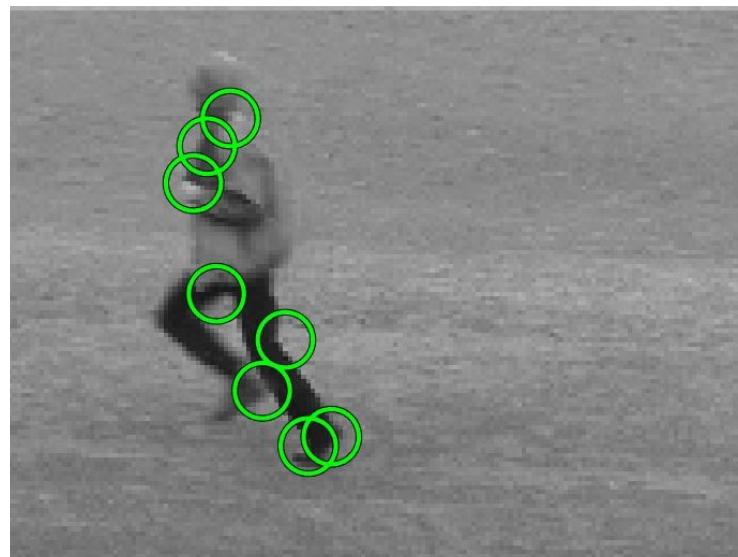


### **Gabor Detector Results :**

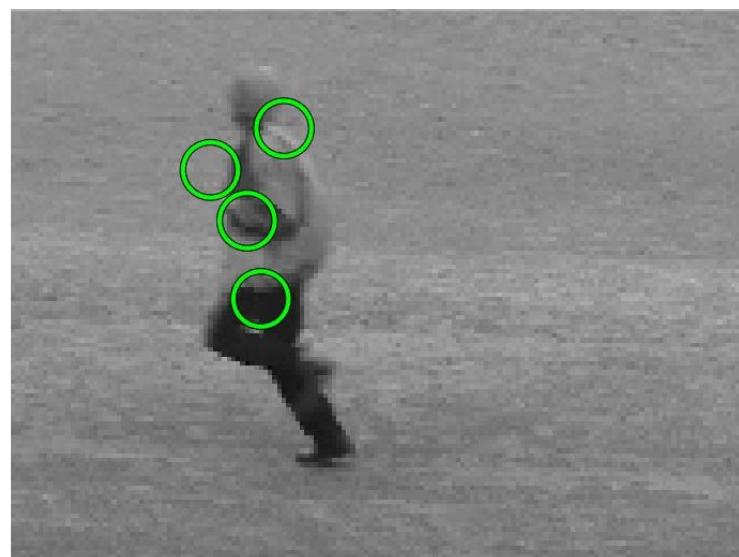


Για το **running** χρησιμοποίησα από video με όνομα **person01\_running\_d3\_uncomp** και συγκεκριμένα επέλεξα να σας δείξω τα frames 16,17,18(από πάνω προς τα κάτω) :

### Harris Detector Results :



**Gabor Detector Results :**



Για το **walking** χρησιμοποίησα από video με όνομα **person11\_walking\_d2\_uncomp** και συγκεκριμένα επέλεξα να σας δείξω τα frames 62,63,64(από πάνω προς τα κάτω) :

### Harris Detector Results :



### **Gabor Detector Results :**



Τα παραπάνω στιγμιότυπα είναι αρκετά κατατοπιστικά όσο αφορά την ορθότητα του τρόπου με τον οποίο υλοποιήσαμε τους 2 detectors, ωστόσο για να είναι προφανές ότι το αποτέλεσμα που παράγουμε είναι σωστό μπορείτε να τρέξετε το script που σας δίνουμε με όνομα **test1\_3.m** το οποίο εμφανίζει για οποιοδήποτε από τα video εσεις επιθυμείτε να ελέγχεται το αποτέλεσμα και των δύο detectors διαδοχικά(το μόνο που καλείστε εσείς να συμπληρώσετε για να τρέξει είναι το όνομα του video που θέλετε να διαβάσετε).

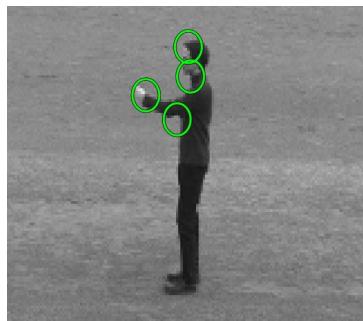
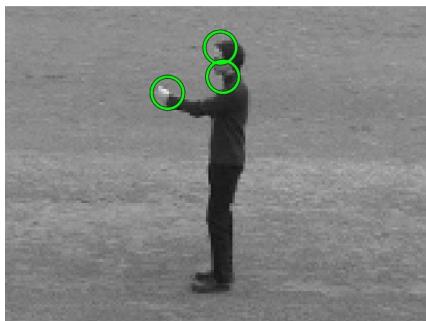
Στη συνέχεια είναι πολύ σημαντικό για την εξαγωγή ασφαλών συμπερασμάτων να πειραματιστούμε με τις τιμές των  $\sigma$  και  $\tau$ . Για το σκόπο αυτό γράψαμε το script με όνομα **ParametersCheck.m** το οποίο αρχικά κάνει load 3 .mat αρχεία στα οποία μέσα είχαμε αποθηκεύσει το αποτέλεσμα του readVideo() για καθένα από τα video που μας δίνεται για να μπορούμε να τα έχουμε άμεσα διαθέσιμα και να μην γράφουμε συνέχεια τις ίδιες εντολές. Στο script αυτό αυτό που στην ουσία κάνουμε είναι να τρέχουμε ενα διπλό loop για καθένα από τα δοσμένα video και να παράγουμε 84 figures για καθένα από τα videos(42 figures για κάθε detector). Τα figures αυτά αποθηκεύονται σε κατάλληλους φακέλους για να μπορούμε να τα επεξεργαστούμε και το όνομα του καθενός υποδηλώνει το συνδυασμό παραμέτρων  $\sigma, \tau$  που έδωσα αυτό το αποτέλεσμα. Προφανώς τα figures είναι πάρα πολλά, τόσο για να τα δείξω όλα, όσο και να τα επισυνάψω στο zip αρχείο αλλά σας δίνω το script **ParametersCheck.m** που κάνει την δουλειά αυτή και στην συνέχεια σας παραθέτω μερικά χαρακτηριστικά παραδείγματα για να σας αναφέρω τα συμπεράσματα τα οποία βγάζω.

Επίσης επειδή δεν μπορούσαμε να αποφανθούμε εύκολα σχετικά με τον τρόπο με τον οποίον επηρεάζουν οι παράμετροι  $\sigma$  και  $\tau$  τον Gabor detector, φτιάξαμε ενα άλλο script με όνομα **GaborTest.m** στο οποίο αρχικά βρίσκουμε το αποτέλεσμα που αντιστοιχεί στην ‘σωστή’ επιλογή παραμέτρων που εμείς βρήκαμε ( $\sigma=2$ ,  $\tau=1.5$ ) και μετά κρατάμε σταθερή την μία παράμετρο και μεταβάλλουμε την άλλη. Αυτό το κάνουμε για όλα τα video που μας δίνεται και τα αποτελέσματα αποθηκεύονται σε κατάλληλα directories για να γίνει σωστός διαχωρισμός. Με τον τρόπο αυτό καταφέραμε να βγάλουμε τα επιθυμιτά αποτελέσματα, κάτι το οποίο θα δείτε στη συνέχεια της αναφοράς (το script **GaborTest.m** σας το επισυνάπτω στο zip αρχείο και θα είναι σε μορφή τέτοια ώστε να τρέχει χωρίς να θέλει κάποια αλλαγή από εσας).

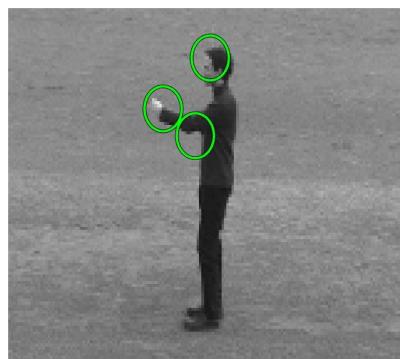
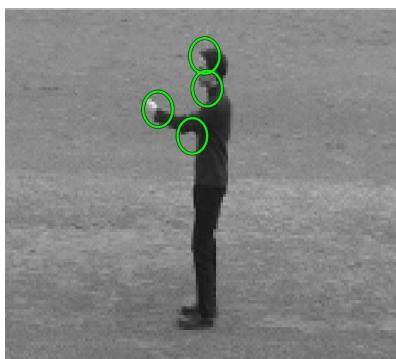
Τα αποτελέσματα φαίνονται στη συνέχεια :

### Harris detector

**Boxing** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.5), (2,0.9), (2,2.1) αντίστοιχα.

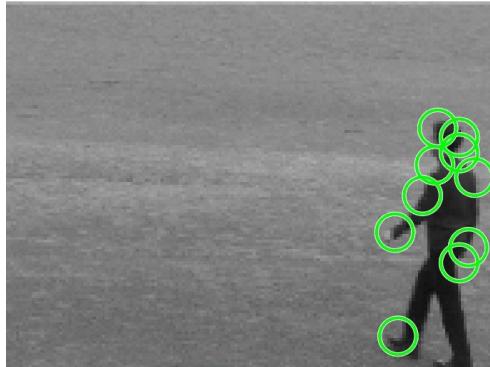


Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.9), (2.5,0.9), (5,0.9) αντίστοιχα.



**Συμπεράσματα** : Συμπεραίνουμε λοιπόν ότι όσο αυξάνουμε το  $\tau$  μπορεί αρχικά να μην χαλάει το αποτέλεσμα που έχουμε αλλά αν η αύξηση αυτή υπερβεί ενα κατώφλι τότε το αποτέλεσμα που παίρνουμε δεν είναι τόσο καλό, όπως φαίνεται και στις παραπάνω εικόνες χαρακτηριστικά, αφού όσο αυξάνουμε το  $\tau$  ουσιαστικά κοιτάμε και σχετικά "παλιές" και πολύ "μεταγενέστερες" στιγμές κάτι το οποίο δεν είναι πάντα επιθυμητό για τα interest points. Επίσης μια μικρή αύξηση του  $\sigma$  δεν χαλάει και πολύ το αποτέλεσμα αλλά αν η αύξηση αυτή είναι αρκετά μεγάλη η ανίχνευση είναι πολύ χοντροκομμένη και δεν δίνει σωστά αποτελέσματα με την έννοια ότι οι κύκλοι πλέον είναι αρκετά μεγάλοι και δεν μπορείς να καταλάβεις σε ποιά σημεία έγινε η ανίχνευση κίνησης, δηλαδή επειδή είναι μεγάλης έκτασης τα σημεία ενδιαφέροντος δε βοηθαίει στο classification που θα κανουμε στη συνεχεία καθώς τετοιες μεγαλες παρομοιες περιοχες μπορει να εντοπιζονται και σε εικονες που δεν ανηκουν στον ίδιο θεματικο κυκλο. Αρα το αποτέλεσμα δεν είναι καλό και σίγουρα δεν είναι αυτό που θα περιμέναμε.

**Walking** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.5), (2,0.9), (2,2.1) αντίστοιχα.

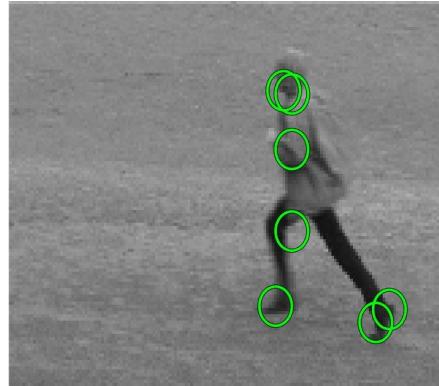
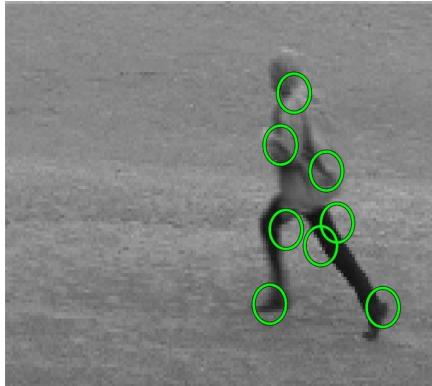


Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.9), (2.5,0.9), (5,0.9) αντίστοιχα.

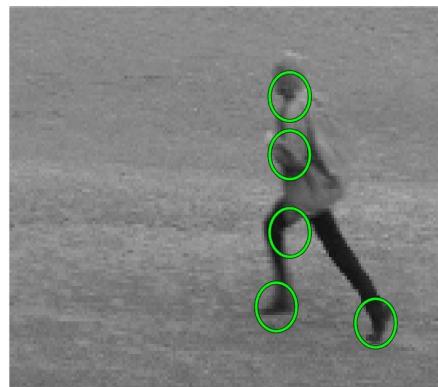
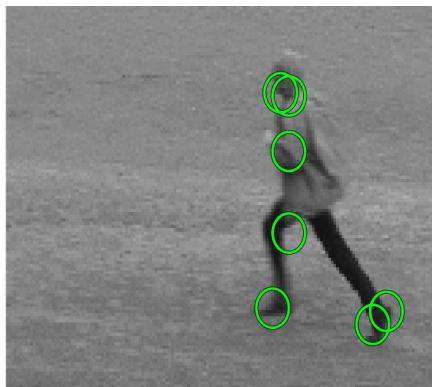


**Συμπεράσματα :** Τα συμπεράσματα τα οποία βγάζουμε για το walking είναι παρόμοια με αυτά τα οποία ανέφερα για το boxing παραπάνω. Αυτό που θα μπορούσαμε να προσθέσουμε εδώ είναι ότι οταν η παράμετρος  $\tau$  υπέρβει το κατώφλι της τότε το αποτέλεσμα χαλάει πάρα πολύ και ουσιαστικά είναι λάθος (σχήμα πάνω δεξιά).

**Running** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.5), (2,0.9), (2,2.1) αντίστοιχα.



Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,0.9), (2.5,0.9), (5,0.9) αντίστοιχα.



**Συμπεράσματα** : Τα συμπεράσματα τα οποία βγάζουμε για το running είναι παρόμοια με αυτά τα οποία ανέφερα για το boxing παραπάνω, αλλα θα μπορούσαμε να παρατηρήσουμε ότι οι μεγάλες αλλαγές σε μια από τις δύο παραμέτρους χαλάνε το αποτέλεσμα αλλά οχι τόσο όσο το έκαναν στις περιπτώσεις του boxing και ειδικά του walking.

### Gabor detector

**Boxing** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (2,1), (2,2) αντίστοιχα.



Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (1,1.5), (3,1.5) αντίστοιχα.



**Συμπεράσματα** : Από τα παραπάνω figures συμπεραίνω ότι αν μειωθεί κατά ενα παράγοντα το  $\tau$  τότε το αποτέλεσμα που θα πάρω θα απέχει αρκετά από το ιδανικό αλλά ανιχνεύει την κίνηση αλλά οχι σε όλα τα σημεία. Αν υπάρξει αύξηση του  $\tau$  τότε η ανίχνευση μας είναι εντελώς λάθος και δεν μπορεί να κριθεί άξια συμπερασμάτων διότι όπως ανέφερα και παραπάνω κοιτάμε και σχετικά "παλιές" και πολύ "μεταγενέστερες" στιγμές κάτι το οποίο δεν είναι παντά επιθυμητό για τα interest points. Αν τώρα κρατήσουμε σταθερό το  $\tau$  και μειώσουμε το  $\sigma$  βλέπουμε ότι μικραίνοντας το scale έχουμε πιο λεπτομερή ανίχνευση και αυξάνονται τα κυκλάκια στην εικόνα κάτι το οποίο μπορεί εύκολα να οδηγήσει σε λάθος ανίχνευση διότι όσο πιο μικρό είναι το κυκλάκι τόσο ακριβέστερος πρέπει να είσαι για την περιοχή της κίνησης και κάτι τέτοιο δεν είναι πάντα εφικτό να γίνει. Αυξάνοντας τώρα το  $\sigma$  η ανίχνευση μας γίνεται πιο 'χοντροκομμένη' και χάνει αρκετή από την ακρίβεια της αλλά δεν μπορεί να ισχυριστούμε ότι οδηγεί σε τελείως λάθος αποτελέσματα αφού ανιχνεύει κάποια σωστά και κάποια λανθασμένα σημεία κίνησης.

**Walking** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (2,1), (2,2) αντίστοιχα.



Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (1,1.5), (3,1.5) αντίστοιχα.



**Συμπεράσματα :** Στη περίπτωση του walking τα παραπάνω συμπεράσματα ισχύουν για την περίπτωση της μεταβολής του  $\sigma$  όπως είναι προφανές και από τα παραπάνω figures. Ωστόσο παρατηρούμε εδώ ότι μια οποιαδήποτε μεταβολή του  $\tau$ , είτε αύξηση είτε μείωση, θα οδηγεί σε τελείως λάθος ανίχνευση, αφού δεν καταφέρνει να εντοπίσει κανένα σημείο κίνησης(οι δύο πιο δεξιά εικόνες στις πάνω εικόνες).

**Running** : Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (2,1), (2,2) αντίστοιχα.



Οι τρεις παρακάτω εικόνες έχουν τιμές για το  $\sigma$  και το  $\tau$  (2,1.5), (1,1.5), (3,1.5) αντίστοιχα.



**Συμπεράσματα :** Στη περίπτωση του running είναι προφανές από τα παραπάνω figures οτι ισχύουν τα ίδια συμπεράσματα με αυτά που ισχύουν στο walking και για το λόγο αυτό δεν τα αναφέρω ξανά.

**Παρατήρηση :** Ο πειραματισμός για **πολλαπλές κλίμακες** έγινε αλλα δεν οδήγησε σε κάποιο αφαλές και άξιο αναφοράς συμπέρασμα. Ωστόσο σας αναφέρω για πληρότητα ποιοτικά τον τρόπο με τον οποίο έγινε ο πολυκλιμακώτος πειραματισμός και έλεγχος : Η υλοποίηση του μπορεί να γίνει με μια for loop, στην οποία κάθε σημείο ενδιαφέροντος κάθε κλίμακας θα μπαίνει στον πίνακα(concat) τον οποίο έχουν δημιουργήσει και επιστρέψει οι άλλες κλίμακες. Με τον απλοικό αυτό τρόπο θα καταφέρουμε να υπολογίσουμε και τα πιο 'χοντρά' και τα πιο 'λεπτά' σημεία ενδιαφέροντος. Στην οπτικοποιήση αυτο που αλλαζει είναι οτι βλέπουμε και μεγάλους και μικρούς κύκλους. Πολλαπλές κλίμακες στο scale του χρονου, ακόμη και με μικρές αλλαγές χαλούσε τα αποτελέσματα καθως εβγαζε σημεια τα οποία δεν ηταν πραγματικα σημεία ενδιαφέροντος.

## Μέρος 2 : Χωρό-χρονικοί Ιστογραφικοί Περιγραφητές

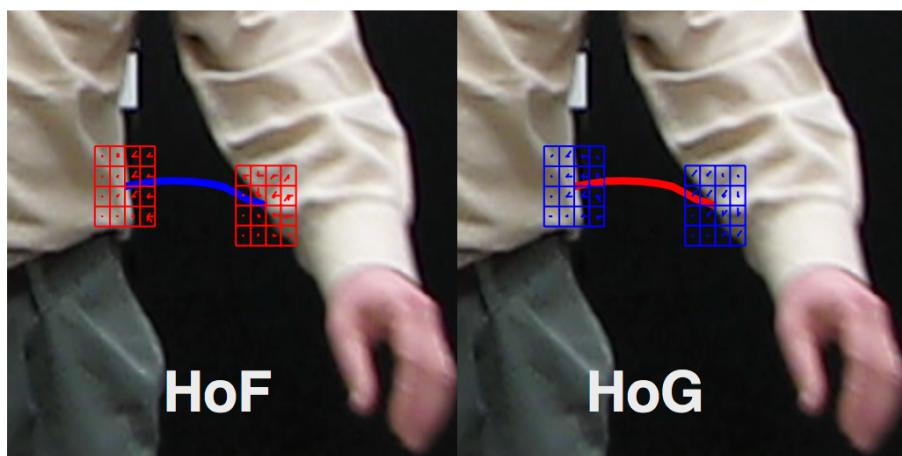
Στο μέρος αυτό, εφόσον έχουμε υπολογίσει τα σημεία ενδιαφέροντος από το Μέρος 1 θα τα χρησιμοποιήσουμε, αφού θα οι χωρό-χρονικοί περιγραφητές που θα χρησιμοποιηθούν βασίζονται στον υπολογισμό ιστογραμμάτων της κατευθυντικής παραγώγου(HOG) και της οπτικής ροής(HOF) γύρω από αυτά τα σημεία ενδιαφέροντος.

**HOG(Histogram Of Oriented Gradients)** : Η βασική ιδέα πίσω από την δημιουργία των HOG περιγραφητών είναι ότι η όψη του τοπικού αντικείμενου(local object appearance) και το σχήμα μέσα σε μια εικόνα μπορεί να περιγραφεί από την κατανομή των κλίσεων έντασης(intensity gradients) ή τις κατευθύνσεις των ακμών. Η εφαρμογή HOG περιγραφητών μπορεί να γίνει χωρίζοντας την εικόνα σε μικρές συνδεδεμένες περιοχές που ονομάζονται κελία και για κάθε κελί υπολογίζεται ενα ιστόγραμμα κατευθύνσεων κλίσης ή ο προσανατολισμός των ακμών για τα pixels εντός του κάθε κελιού. Ο συνδυασμός αυτών των ιστογραμμάτων ουσιαστικά αντιπροσωπεύει τον detector. Για πιο βελτιωμένες αποδόσεις θα μπορούσε να γίνει σε καθένα από τα ‘τοπικά’ ιστογράμματα ενα contrast-normalization.

**HOF(Histogram of Optical Flow)** : Η μέθοδος αυτή βασίζεται στην εξαγωγή χαρακτηριστικών κίνησης από εικόνες χρησιμοποιώντας την οπτική ροή. Το βασικό πλεονέκτημα της μεθόδου αυτής είναι ότι η ευθύνη για σωστή εκτίμηση κίνησης περιορίζεται στον απλό υπολογισμό της οπτικής ροής. Θα πρέπει να σημειωθεί ότι υπάρχουν πολλές προσεγγίσεις για τον υπολογισμό της οπτικής ροής. Οι HOF descriptors είναι πιο ’ακριβή’ για την εξαγωγή για των χαρακτηριστικών σε σχέση με τους HOG και για τον λόγο αυτό οι HOG χρησιμοποιούνται περισσότερο.

Ενα παράδειγμα που δείχνει μια χαρακτηριστική διαφορά των HOG και HOF περιγραφητών φαίνεται παρακάτω :

**HoG** → local appearance at each frame  
**HoF** → local motion at each frame



Μετά την θεωρητική παρουσίασει των HOG και HOF περιγραφητών παρουσιάζουμε συνοπτικά την πορεία που ακολουθήσαμε για την υλοποίηση του ερωτήματος :

Αρχικά για κάθε frame (200 συνολικά) του κάθε video υπολογίσαμε το gradient(κατευθυντική παράγωγο) με χρήση της συνάρτησης **imgradient()** του Matlab καθώς και την οπτική ροή με χρήση της συνάρτησης **Ik()** που είχαμε υλοποιήσει στη 2<sup>η</sup> εργαστηριακή άσκηση. Τα αποτελέσματα που μας έδωσαν αυτές οι συναρτήσεις τα κρατήσαμε σε **cell arrays** ώστε να τα έχουμε όλα διαθέσιμα στις παρακάτω ενέργειές μας προκείμενου να υπολογίσουμε τους HOG (για το gradient) και HOF (για το optical flow) descriptors.

**Παρατήρηση:** Το Optical Flow αφού υπολογίζεται σε σχέση με 2 pixels υπολογίστηκε σε πίνακα με 199 Τρίτη διάσταση (το τελευταίο frame δεν έχει 201<sup>o</sup> ώστε να υπολογιστεί το optical flow του). Πιο απλά το Opt(i) είναι το αποτέλεσμα της lk για το i-οστο και i-οστο+1 frame κάθε βίντεο.

Στη συνέχεια(ερώτημα 2.2), με βάση το διανυσματικό πεδίο που υπολογίσαμε από πάνω (κατευθυντικές παραγώγους ή optical flow) θα εξάγουμε τον κατάλληλο περιγραφητή. Για να το επιτύχουμε αυτό κάνουμε χρήση της συνάρτησης **OrientationHistogram.p**. Στον κώδικα μας ο υπολογισμός κάθε περιγραφητή γίνεται στη συνάρτηση **Descriptors()**. Εκείνη παίρνει σαν όρισμα τα σημεία ενδιαφέροντος που βρήκε η κάθε μέθοδος του 1<sup>ου</sup> ερωτήματος (υπολογίζουμε περιγραφητές τόσο για τα σημεία που βρήκε η μέθοδος Harris όσο και η μέθοδος Gabor), την κλίμακα που χρησιμοποιήθηκε κατά τον εντοπισμό αυτών των σημείων ενδιαφέροντος σε κάθε μία από τις προαναφερθείσες μεθόδους, τις τιμές που μας έδωσε η **imgradient()** και η **Ik()** για κάθε frame κάθε ενός από τα βίντεο (η κλήση της **Descriptors()** γίνεται 1 φορά για κάθε ένα βίντεο, σύνολο 9 φορές), καθώς και τις διαστάσεις του κάθε βίντεο. Αφού τα λάβει όλα αυτά καλεί για κάθε σημείο ενδιαφέροντος την **OrientationHistogram.p** με πλήθος nbins=8 και μέγεθος grid [n m] = [4 4] και υπολογίζει για κάθε τέτοιο σημείο ενδιαφέροντος τη γειτονιά του που έιναι τετραγωνική με μήκος πλευράς 8\*scale (αυτή την κλίμακα που δώσαμε κατά την κληση της **Descriptors()**) και με βάση αυτή υπολογίζει το κατάλληλο ιστόγραμμα, το βάζει σε έναν array τον οποίο επιστρέφει (και το script αναλαμβανει να το βάλει στην κατάλληλη θέση του cell array που θα περιέχει μετά την εκτέλεση της loop αυτής όλα τα ιστογράμματα όλων των περιγραφητών για όλα τα ενδιαφέροντα σημεία όλων των βίντεο). Αν τυχόν δεν υπάρχει κατάλληλη τέτοια γειτονιά επείδή το σημείο ενδιαφέροντος κάποιου frame βρίσκεται σε οριακή θέση, τότε παίρνουμε τη γειτονιά του κανονικά σε όποια πλευρά μπορούμε και έως τα όρια της εικόνας, σε όσες πλευρές δεν είναι δυνατό να επεκταθούμε λόγω των ορίων του frame. Αυτό επιτυγχάνεται με τις if που υπάρχουν στον κώδικα του **Descriptors()**. Επίσης, σημειώνουμε ότι αν βρεθεί σημείο ενδιαφέροντος στο 200<sup>st</sup> frame δε μπορούμε να υπολογίσουμε το ιστόγραμμα για το HOF καθώς, όπως προείπαμε δεν έχουμε optical flow

σε αυτό το καρέ (τελευταίο) κάθε βίντεο. Το μέγεθος κάθε περιγραφητή, όπως αναφέρεται και στην εκφώνηση που εξάγει η **OrientationHistogram.p** έιναι  $n \times m \times nbins$ .

Άρα μετά το πέρας αυτή της loop υπολογισμού των περιγραφητών και για τα 9 βίντεο(είναι λίγο χρονοβόρο λόγω του υπολογισμού της οπτικής ροής σε κάθε frame κάθε βίντεο) έχουμε σε cell arrays τους HOG και HOF περιγραφητές, τόσο για τα σημεία ενδιαφέροντος που υπολογίστηκαν με τη μέθοδο Gabor όσο και με τη μέθοδο Harris(τα ονόματα των μεταβλητών στον κώδικα κάνουν ξεκάθαρο ποιος cell array περιέχει τι).

Τέλος(ερώτημα 2.3) θα υπολογίσουμε την τελική αναπαράσταση(global representation) για κάθε βίντεο υλοποιώντας την bag of visual words (BoVW) τεχνική όπως αυτή περιγράφεται στην εκφώνηση της 1<sup>ης</sup> εργαστηριακής άσκησης. Στην περίπτωσή μας, η διαφορά που υπάρχει σε σχέση με όσα περιγράφονται στην 1<sup>η</sup> εργαστηριακή άσκηση είναι ότι εδώ δεν έχουμε δεδομένα test και train αλλά μόνο δεδομένα (αυτά που μας έδωσαν οι περιγραφητές) με βάση τα οποία θα υπολογιστούν αρχικά οι λέξεις του λεξικού και στη συνέχεια τα ιστογράμματα εμφάνισης. Για να το επιτύχουμε αυτό λοιπόν δημιουργήσαμε μια συνάρτηση **MyBagOfWords.m** παρόμοια με αυτή που φτιάξαμε για το προαιρετικό ερώτημα της 1<sup>ης</sup> εργαστηριακής άσκησης, απλά αφαιρώντας τα sections του κώδικα που ασχολούνται με το data\_test κομμάτι των δεδομένων. Στη συνέχεια του script μας λοιπόν, καλούμε αυτή τη συνάρτηση, η λειτουργία της οποίας παρατίθεται παρακάτω, για κάθε περιγραφητή (HOG,HOF,HOG/HOF, ο HOG/HOF περιγραφητής υπολογίζεται συνενώνοντας τους περιγραφητές HOG και HOF σε έναν array) για κάθε μέθοδο υπολογισμού σημείων ενδιαφέροντος(Gabor,Harris). Τα αποτελέσματα που μας δίνει η συνάρτηση **MyBagOfWords.m** κρατιούνται σε έναν array, το όνομα του οποίου στον κώδικα κάνει ξεκάθαρο τον τύπον του περιγραφητή που χρησιμοποιήθηκε για τη δημιουργία κάθε ιστογράμματος εμφάνισης.

Λεπτομέρεις για την **MyBagOfWords.m**:

- Η **MyBagOfWords.m** παίρνει ως ορίσματα τον cell array που προέκυψε, τον data\_train.
- Ορίζουμε έναν αριθμό κέντρων για τον οποίο θα τρέξει ο αλγόριθμος kmeans. Αυτός ο αριθμός κυμαίνεται από 20-50.
- Στη συνέχεια για να κάνουμε συνένωση των περιγραφητών του συνόλου εκπαίδευσης (train) σε ένα ενιαίο διάνυσμα χαρακτηριστικών ανεξάρτητα από την κλάση που ανήκουν, παίρνουμε τον ανάστροφο του cell array data\_train ώστε χρησιμοποιώντας τη συνάρτηση cell2mat να πάρουμε αυτό που θέλουμε. (κάθε εσωτερικός πίνακας του cell array εχει μεταβλητό αριθμό γραμμών αλλά σταθερό αριθμό στηλών, έτσι παίρνωντας τον ανάστροφο προκύπτει ένας πίνακας με όλες τις γραμμές των πινάκων του cell array data\_train τη μία κάτω από την άλλη).
- Στη συνέχεια επιλέγουμε τυχαία το 50% αυτών των γραμμών (από το διάνυσμα των χαρακτηριστικών) για την υλοποίηση του αλγορίθμου kmeans όπως αναφέρεται και στην εκφώνηση της 1<sup>ης</sup> εργαστηριακής άσκησης..

- Έπειτα με βάση αυτά που επιστρέφει η kmeans υπολογίζουμε για κάθε περιγραφητή την ελάχιστη ευκλείδια απόσταση από τα κέντρα που μας έδωσε το kmeans.
- Για τις αποστάσεις αυτές βρίσκουμε ποιο κέντρο κάθε φορά μας την έδωσε και ανξάνουμε ανάλογα την αντίστοιχη θέση του πίνακα BF\_tr (για το data\_train) υπολογίζοντας έτσι στην ουσία χειροκίνητα το ιστόγραμμα (χωρίς χρήση της histc).
- Αφού το υπολογίσουμε αυτό, κανονικοποιούμε με τη L2 νόρμα όπως υποδεικνύεται και επιστρέφουμε αυτές τις κανονικοποιημένες τιμές.

## Μέρος 3 : Κατασκευή Δενδρογράμματος για τον Διαχωρισμό Δράσεων

Στο τελευταίο ερώτημα της άσκησης, θα χρησιμοποιήσουμε τις BoVW αναπαραστάσεις που υλοποιήσαμε στα προηγούμενα ερωτήματα και θα προσπαθήσουμε να κατηγοριοποιήσουμε τα video με τις ανθρώπινες δράσεις σε 3 κλάσεις. Η προσπάθεια αυτή θα επιτευχθεί με την οπτικοποίηση των διανυσμάτων χαρακτηριστικών μέσω κατασκευής δενδρογράμματος αποστάσεων το οποίο αντιπροσωπεύει την ικανότητα διαχωρισμού των 3 διαφορετικών κατηγοριών.

Για την κατασκευή του δενδρογράμματος αποστάσεων από το σύνολο των BoVW ιστογραμμάτων χρησιμοποιήσαμε την  $\chi^2$  απόσταση της όποιας ο μαθηματικός τύπος δίνεται παρακάτω:

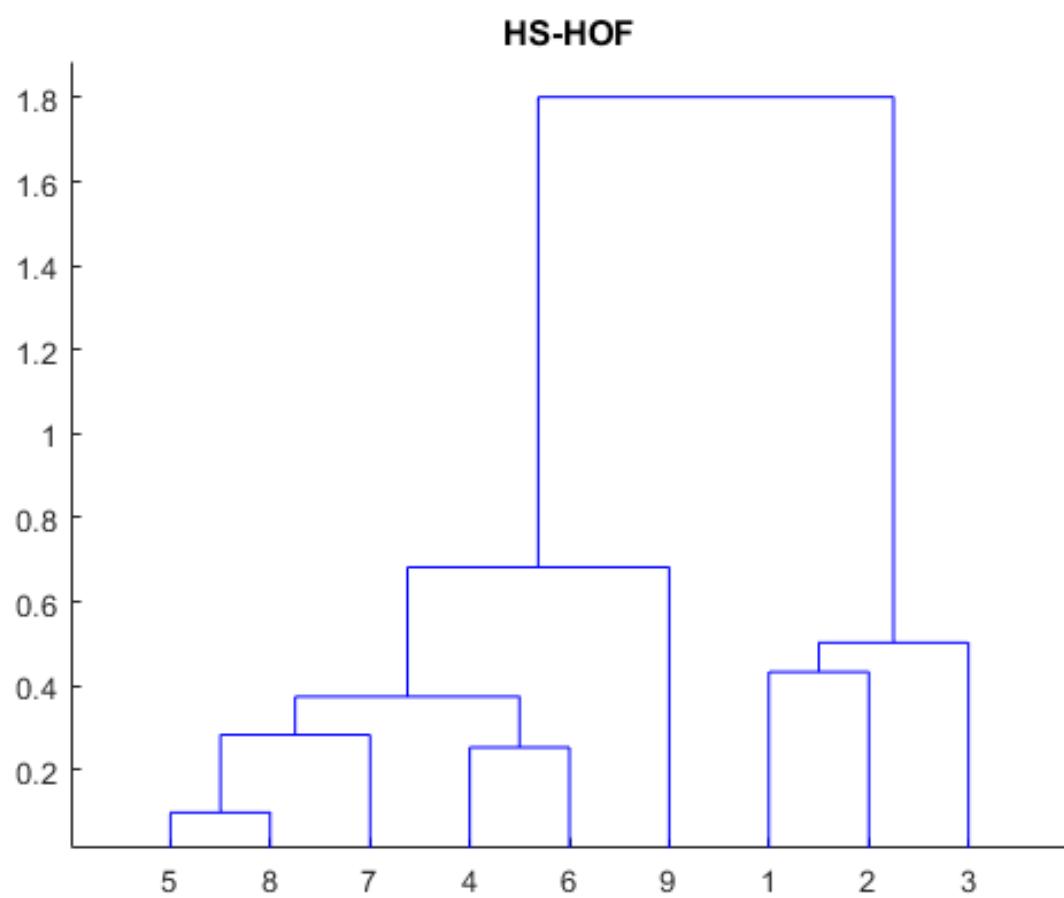
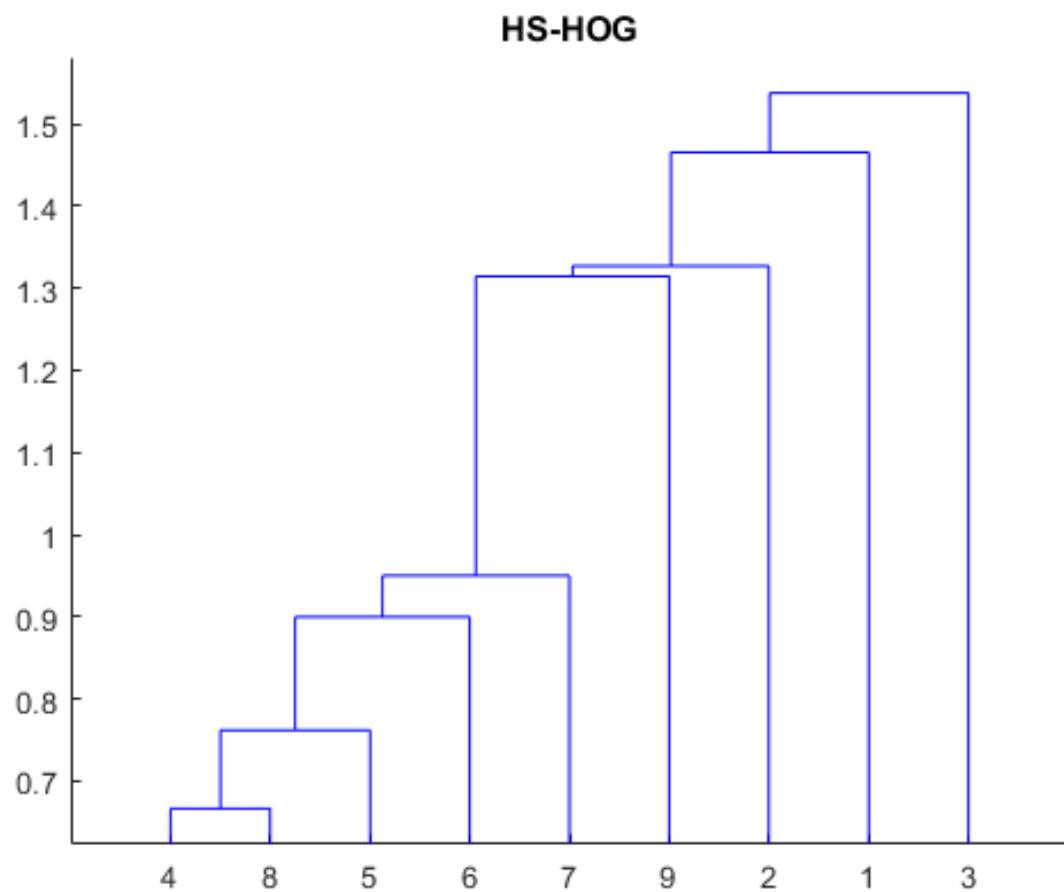
$$D(H_i, H_j) = \frac{1}{2} \sum_{n=1}^K \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}}$$

Η κατασκευή των δενδρογραμμάτων έγινε με πολλαπλή εκτέλεση των παρακάτω εντολών :

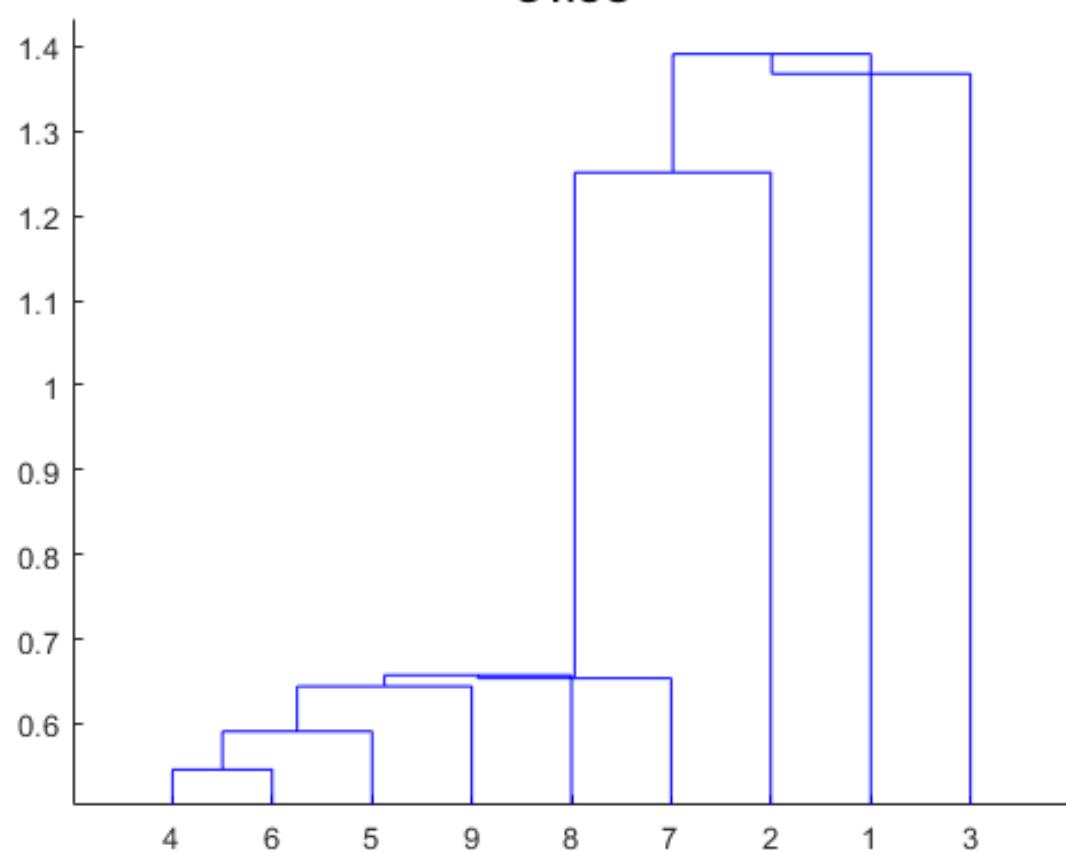
```
link_<'descriptor_type'> = linkage(<'output of bag of words for that descriptor_type'>, metric', '@distChiSq');
dendrogram(link_<'descriptor_type'>);
```

Πειραματιστήκαμε με τις μετρικες της συνάρτησης linkage. Παραθέτουμε τα δενδρογράμματα για μετρική “centroid” και “single” για όλους τους συνδυασμούς detector-descriptor(6 στο σύνολο, HOG,HOF,HOG/HOF για Gabor και Harris-Stephens μέθοδο):

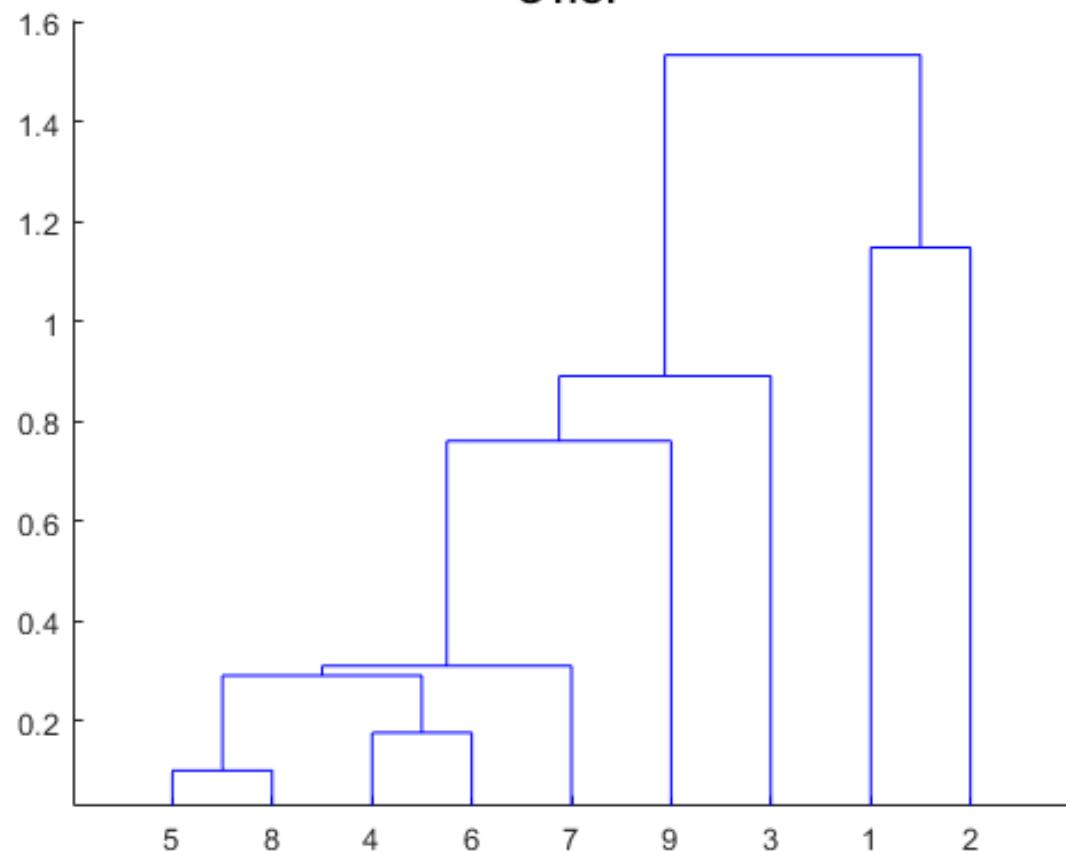
### Centroid Results:



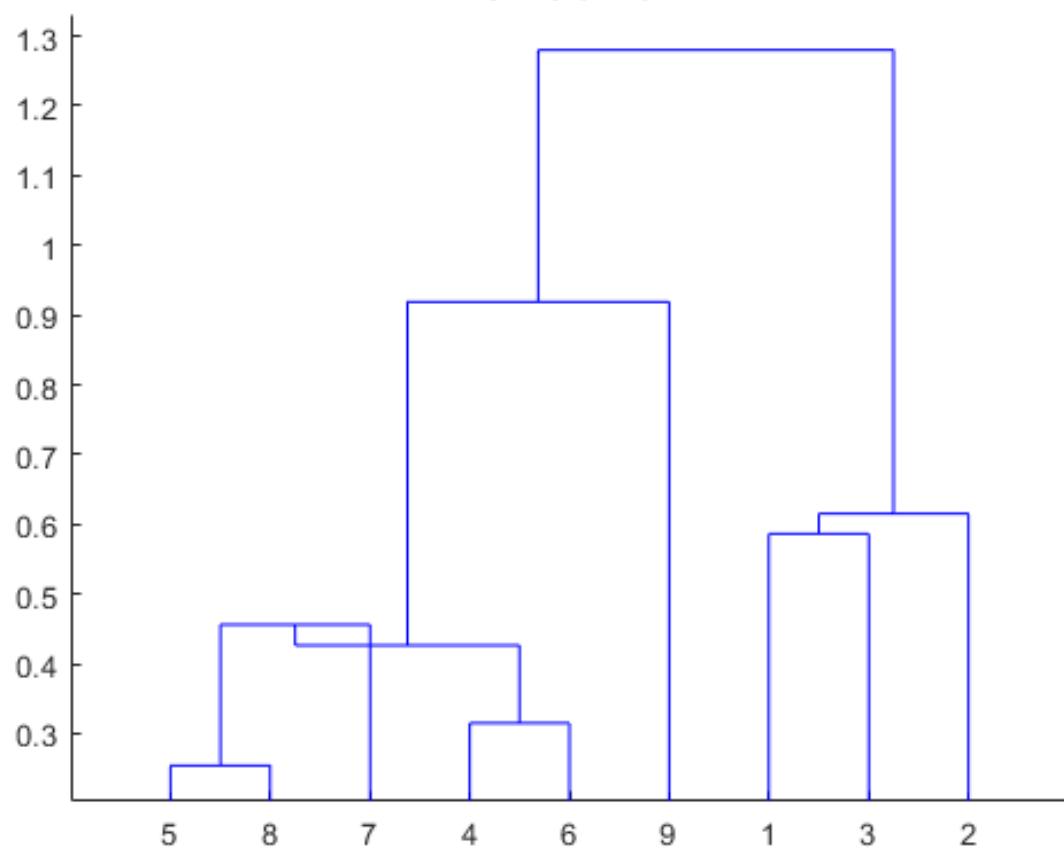
**G-HOG**



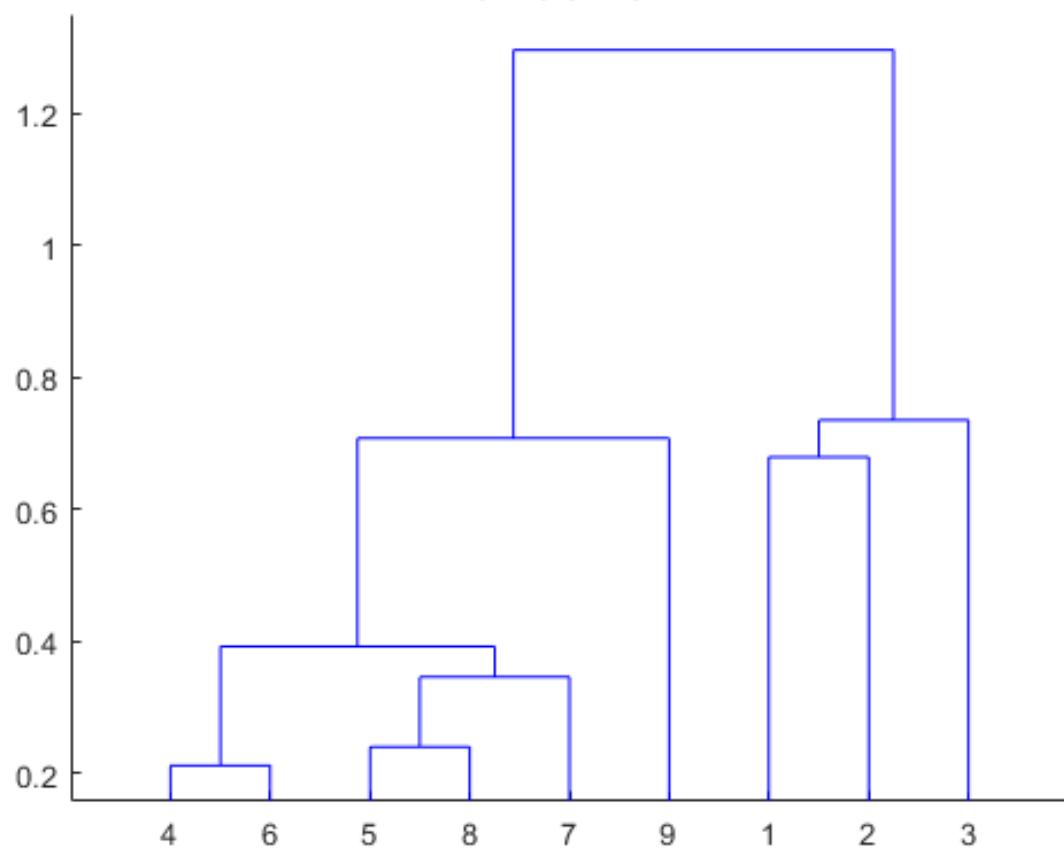
**G-HOF**



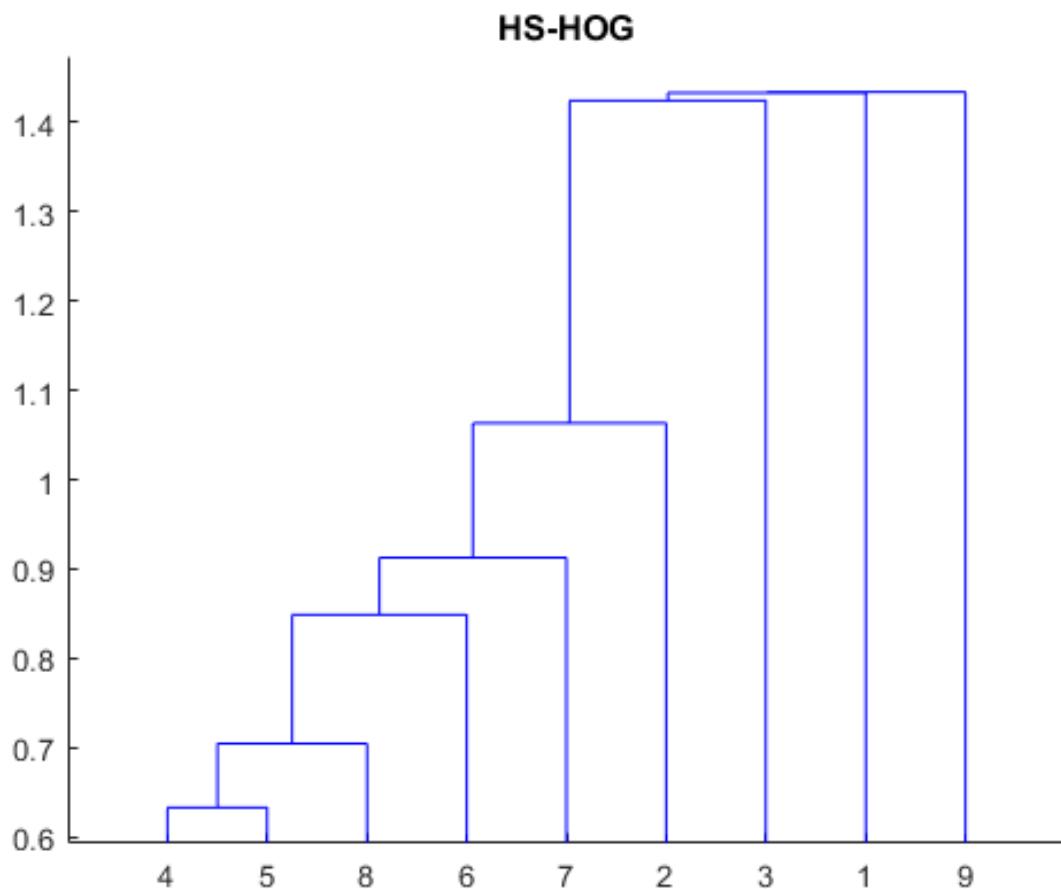
**HS-HOG-HOF**



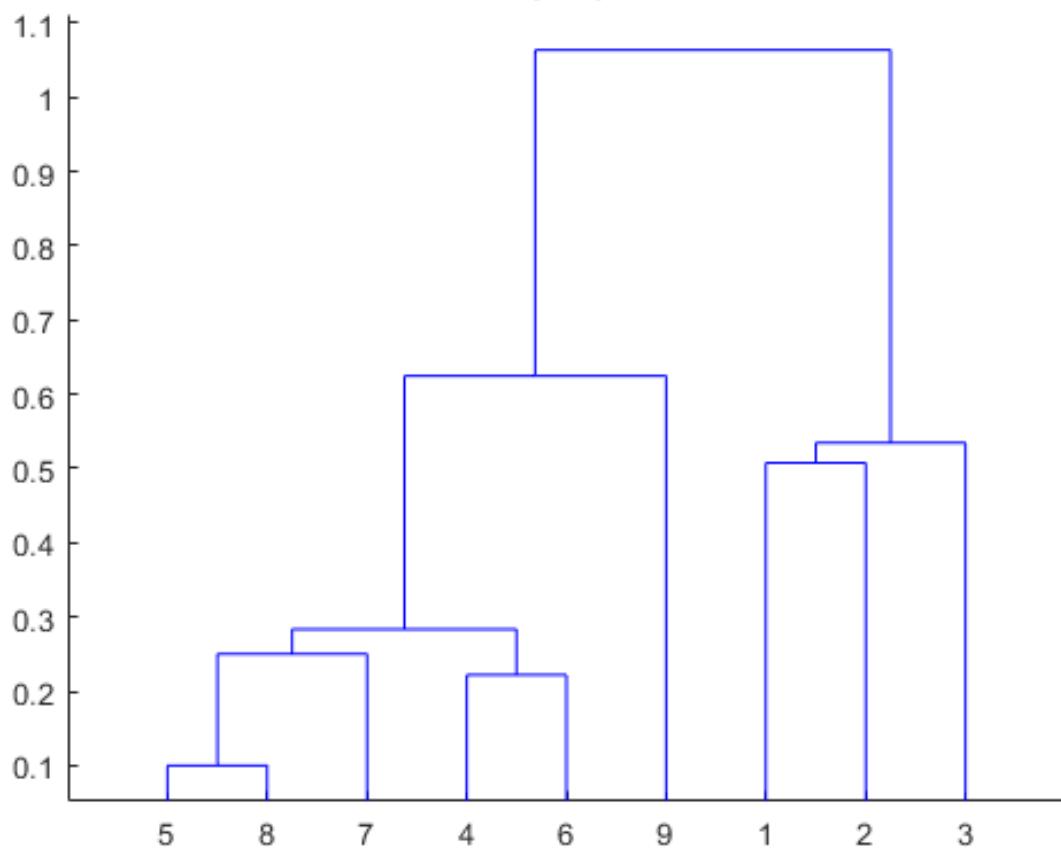
**G-HOG-HOF**



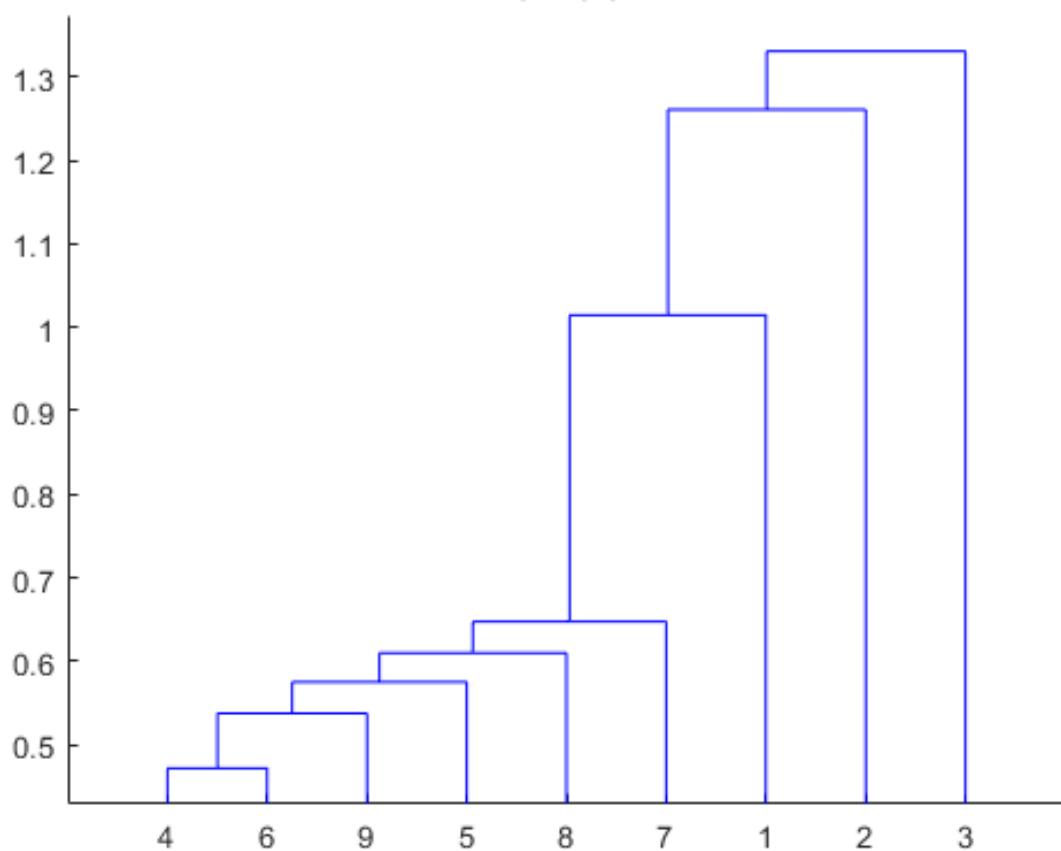
### Single Results:



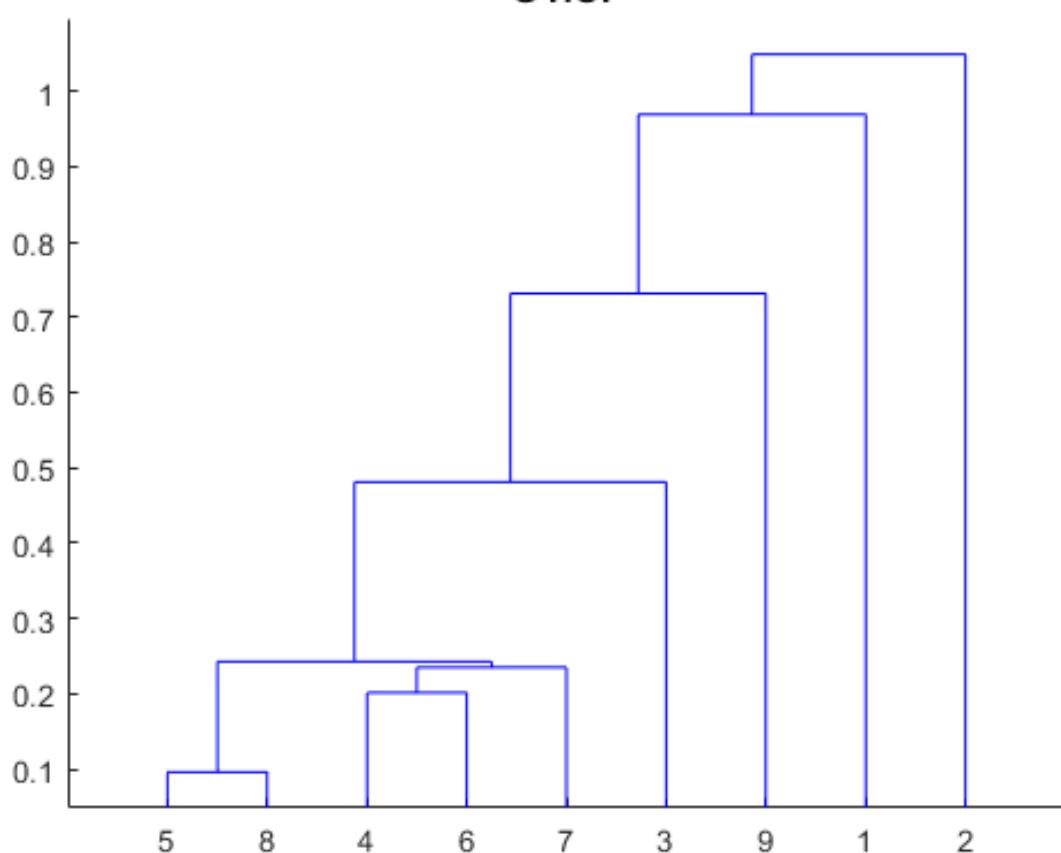
**HS-HOF**



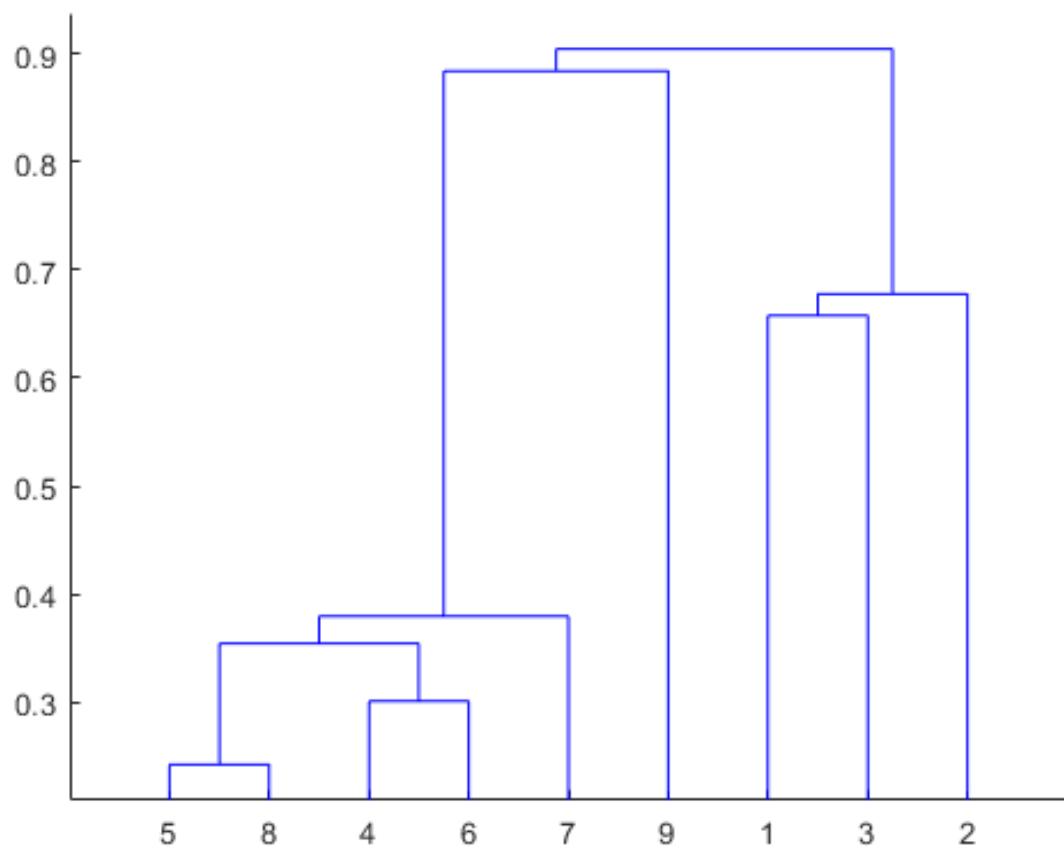
**G-HOG**



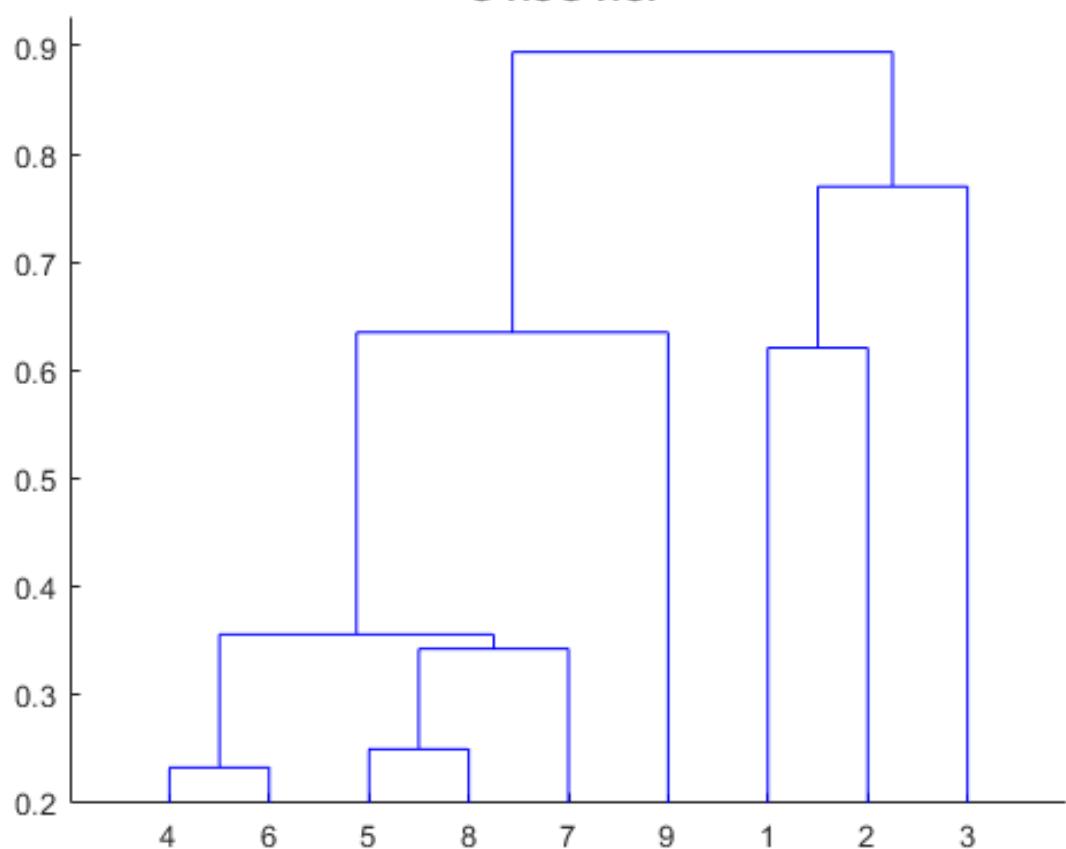
**G-HOF**



**HS-HOG-HOF**



**G-HOG-HOF**



**Σχολιασμός :** Από τα παραπάνω μπορούμε να δούμε ότι η κατηγοριοποίηση με βάση τον περιγραφητή που χρησιμοποιούμε αλλά καμιά φορά και τη μέθοδο είναι περισσότερη ή λιγότερο επιτυχημένη. Γενικά , παρατηρούμε και για τους 2 interest point detectors ότι με τον HOG/HOF περιγραφητή επιτυγχάνουμε καλύτερο classification των video. Χωρίζει το 1,2,3 που αντιστοιχούν σε Boxing από τα υπόλοιπα που αντιστοιχούν σε walking και running. Ωστόσο, επειδή οι 2 αυτές ανθρώπινες δράσεις είναι πανομοιότυπες και γι' αυτό δε γίνεται πλήρης διαχωρισμός (4,5,6/7,8,9).Είναι δηλαδή λίγο μπλεγμένα τα βίντεο αυτά. Ωστόσο η κατηγοριοποίηση είναι σε γενικές γραμμές επιτυχής. Όσον αφορά τον HOG περιγραφητή παρατηρούμε σε κάθε περίπτωση ότι δεν επιτυγχάνει την κατηγοριοποίηση των video και αυτό οφείλεται στο γεγονός ότι αυτός εξάγεται από στατικές πληροφορίες των εικόνων του video. Δεν έχει με άλλα λόγια πληροφορίες για την κίνηση, επομένως η εσφαλμένη κατηγοριοποίηση δεν πρέπει να αποτελεί έκπληξη για εμάς. Τέλος, ο περιγραφητής HOF είναι αρκετά καλύτερος από τον HOG ,καθώς βασίζεται στον υπολογισμό του optical flow και έτσι έχει πληροφορίες για την κίνηση. Κάποιες φορές επιτυγχάνει πολύ καλό classification (πχ βλ. HS-HOF για metric ‘centroid’ και ‘single’, όπου διαχωρίζει τα Boxing videos από τα walking και running) ενώ άλλες δεν είναι τόσο επιτυχημένος(πχ βλ. G-HOF όπου διαχωρίζει 2 από τα 3 videos του boxing σε σχέση με τα άλλα, αφήνοντας τα υπόλοιπα όμως χωρίς σαφή διαχωρισμό , όπως πχ κανει ο HOG/HOF περιγραφητής). Επομένως, λαμβάνοντας υπόψη όλα τα παραπάνω, μπορούμε να συμπεράνουμε ότι αν συνενώσουμε τους περιγραφητές HOG και HOF (αν πάρουμε δηλαδή τον HOG/HOF) έχουμε καλύτερη κατηγοριοποίηση των video μας σε κλάσεις.