

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ



ΟΡΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

(2020-2021)

2^ο Εργαστηριακό Project

*Θέμα: Εκτίμηση Οπτικής Ροής (Optical Flow) και Εξαγωγή Χαρακτηριστικών σε Βίντεο για
Αναγνώριση Δράσεων*

Ονοματεπώνυμο:

- Χρήστος Τσούφης

Αριθμός Μητρώου:

- 03117176

Στοιχεία Επικοινωνίας:

- el17176@mail.ntua.gr

Στόχος της Εργασίας

Το παρόν project αποτελείται από 2 μέρη. Το Μέρος 1 έχει ως στόχο την υλοποίηση ενός συστήματος παρακολούθησης προσώπου και χεριών (Face and Hands Tracking) σε μια ακολουθία βίντεο νοηματικής γλώσσας. Το σύστημα αρχικά θα ανιχνεύει στο πρώτο πλαίσιο την περιοχή του προσώπου και των χεριών (περιοχές ενδιαφέροντος) με χρήση ενός πιθανοτικού ανιχνευτή ανθρώπινου δέρματος. Στη συνέχεια, θα μπορεί να παρακολουθεί τις περιοχές ενδιαφέροντος χρησιμοποιώντας τα εξαγόμενα διανύσματα οπτικής ροής, υπολογισμένα με τη μέθοδο των Lucas-Kanade. Συγκεκριμένα, γίνεται ανίχνευση δέρματος προσώπου και χεριών (1.1), παρακολούθηση προσώπου και χεριών (1.2) [υλοποίηση του αλγορίθμου των Lucas-Kanade (1.2.1), υπολογισμός της μετατόπισης των παραθύρων από τα διανύσματα οπτικής ροής (1.2.2) & πολύ-κλιμακωτός υπολογισμός οπτικής ροής (1.2.3)]. Έπειτα, στο Μέρος 2, μελετάται ο εντοπισμός χωρο-χρονικών σημείων ενδιαφέροντος και η εξαγωγή χαρακτηριστικών σε βίντεο ανθρωπίνων δράσεων. Τα τοπικά χαρακτηριστικά έχουνε δείξει τεράστια επιτυχία σε διάφορα προβλήματα αναγνώρισης της Όρασης Υπολογιστών, όπως η αναγνώριση αντικειμένων. Οι τοπικές αναπαραστάσεις περιγράφουν το προς παρατήρηση αντικείμενο με μια σειρά από τοπικούς περιγραφητές που υπολογίζονται σε γειτονιές ανιχνευθέντων σημείων ενδιαφέροντος. Τελικά, η συλλογή των τοπικών χαρακτηριστικών ενσωματώνεται σε μια τελική αναπαράσταση (global representation), όπως η γνωστή ‘bag of visual words’, ικανή να αναπαραστήσει τη στατιστική κατανομή τους και να προχωρήσει στα επόμενα στάδια της αναγνώρισης. Η αναπαράσταση με χρήση τοπικών χαρακτηριστικών έχει επικρατήσει και στην αναγνώριση ανθρωπίνων δράσεων, όπου γίνεται μια επιλογή από δεδομένα που αφ’ ενός μειώνουν κατά πολύ τη διάσπαση των βίντεο και αφ’ ετέρου τα μετασχηματίζουν σε μια αναπαράσταση που τα κάνει διαχωρίσιμα. Ειδικότερα, εξετάζονται τα χωρο-χρονικά σημεία ενδιαφέροντος (2.1), χωρο-χρονικοί ιστογραφικοί περιγραφητές (2.2) και τέλος, η κατασκευή Bag of Visual Words και η χρήση Support Vector Machines για την ταξινόμηση δράσεων (2.3).

Τεχνολογίες & Τρόπος Εκτέλεσης εφαρμογής

Η παρούσα εργασία υλοποιήθηκε σε ένα Python περιβάλλον και το σετάρισμα της εφαρμογής έγινε σε local περιβάλλον. Οι versions που χρησιμοποιήθηκαν, μετά από την εκτέλεση των παρακάτω εντολών στο terminal είναι:

```
python --version → 3.9.2
```

```
python → import cv2 → cv2.__version__ → 4.5.3
```

```
python → import numpy → numpy.version.version → 1.21.2
```

```
pip3 list | findstr scikit → scikit-image = 0.18.2
```

```
python → import matplotlib → print(matplotlib.__version__) → 3.4.3
```

```
conda → conda -V → conda 4.10.3
```

(Τα πακέτα tqdm, jupyter, nb_conda_kernels είναι fixed)

Το project έχει την εξής δομή: Αποτελείται από 2 ξεχωριστά αρχεία (part1.py, part2.py), ένα για κάθε μέρος της εργασίας, με αντίστοιχα ονόματα. Η εκτέλεση των αρχείων γίνεται μέσω του terminal αφού πρώτα τοποθετηθούν στον ίδιο φάκελο τα κατάλληλα αρχεία που θα χρησιμοποιηθούν ως input.

Μέρος 1: Παρακολούθηση Προσώπου και Χεριών με Χρήση της Μεθόδου Οπτικής Ροής των Lucas-Kanade

1.1 Ανίχνευση Δέρματος Προσώπου και Χεριών

Στο πρώτο ερώτημα, ζητείται η ανίχνευση σημείων δέρματος στο πρώτο πλαίσιο της ακολουθίας και η τελική επιλογή της περιοχής του προσώπου και των χεριών. Για την ανίχνευση των σημείων δέρματος χρησιμοποιείται ο χρωματικός χώρος YCbCr, αφαιρώντας την πληροφορία της φωτεινότητας Y και διατηρώντας τα κανάλια Cb και Cr που περιγράφουν την ταυτότητα του χρώματος. Το χρώμα δέρματος μοντελοποιείται με μια διδιάστατη Γκαουσιανή κατανομή:

$$P(c = \text{skin}) = \frac{1}{\sqrt{|\Sigma|} (2\pi)^2} e^{-\frac{1}{2}(c-\mu)\Sigma^{-1}(c-\mu)'} \quad (1)$$

Όπου c είναι το διάνυσμα τιμών Cb και Cr για κάθε σημείο (x, y) της εικόνας. Η Γκαουσιανή κατανομή εκπαιδεύεται υπολογίζοντας το 2×1 διάνυσμα μέσης τιμής $\mu = [\mu_{Cb}, \mu_{Cr}]^T$ και τον 2×2 πίνακα συνδιακύμανσης Σ από τα δείγματα δέρματος που δίνονται στο αρχείο `skinSamplesRGB.mat` σε μορφή RGB. Η δυαδική εικόνα ανίχνευσης δέρματος προκύπτει από την εικόνα πιθανοτήτων $\Pr(c(x,y)=\text{skin})$, $V(x, y)$ με κατωφλιοποίηση. Ενδεικτικές τιμές κατωφλίου: στο διάστημα $[0.05, 0.25]$ (για τιμές πιθανοτήτων $[0, 1]$).

Για την τελική ανίχνευση των περιοχών δέρματος του προσώπου και των χεριών απαιτείται μια μορφολογική επεξεργασία της δυαδικής εικόνας δέρματος. Συγκεκριμένα, θα γίνει κάλυψη των τρυπών που εμφανίζονται, εφαρμόζοντας `opening` με ένα πολύ μικρό δομικό στοιχείο και `closing` με ένα μεγάλο δομικό στοιχείο. Έτσι, θα εξαλειφθούν οι μικρές περιοχές και θα αποκτήσουν συνοχή οι περιοχές του προσώπου και των χεριών. Τέλος, θα δημιουργηθούν τρία ορθογώνια που θα περιβάλλουν τις περιοχές ενδιαφέροντος-δέρματος (`bounding boxes`) και θα χρησιμοποιηθούν στο 1.2 για τον υπολογισμό των διανυσμάτων Οπτικής Ροής και την τελική παρακολούθηση του προσώπου και των χεριών.

Ζητείται να υλοποιηθεί η παραπάνω διαδικασία σε Python ως αυτόνομη συνάρτηση που να δέχεται ως εισόδους μια εικόνα (την πρώτη της ακολουθίας βίντεο), τη μέση τιμή μ και τη συνδιακύμανση Σ της Γκαουσιανής κατανομής και να επιστρέφει το πλαίσιο οριοθέτησης της περιοχής ενδιαφέροντος στη μορφή `[x, y, width, height]`. Όπου x, y οι συντεταγμένες του παραπάνω αριστερά σημείου, π.χ. `boundingBox = fd(I, mu, cov)`

➤ Βοήθεια για Python: συναρτήσεις `(scipy.stats) multivariate_normal, (cv2) morphologyEx, (scipy.ndimage) label`.

Για να είναι εφικτή η ανίχνευση του ανθρώπινου δέρματος προσώπου και χεριών απαιτείται η στατιστική περιγραφή του χρώματος του δέρματος στην εκάστοτε εικόνα. Επειδή ο RGB χώρος δεν περιέχει άμεσα χρήσιμη πληροφορία για το δέρμα, προτιμάται ο YCbCr χώρος διότι είναι ένας χώρος χρώματος που μπορεί να αναπαρασταθεί το ανθρώπινο δέρμα καθώς το Y είναι η γκριζα φωτεινότητα και τα Cb, Cr οι χρωματικές συνιστώσες. Ο χώρος αυτός προκύπτει από τον M/Σ:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16 & -0.33 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Εάν αφαιρεθεί η πληροφορία της φωτεινότητας Y αλλά παραμείνουν τα Cb & Cr κανάλια που αποτελούν τις χρωματικές συνιστώσες τότε η ανίχνευση των σημείων του δέρματος είναι εφικτή.

Συγκεκριμένα, το χρώμα του δέρματος μοντελοποιείται με μια 2Δ Gaussian Distribution ως εξής:

$$P(c = \text{skin}) = \frac{1}{\sqrt{|\Sigma|} (2\pi)^2} e^{-\frac{1}{2}(\mathbf{c}-\boldsymbol{\mu})\Sigma^{-1}(\mathbf{c}-\boldsymbol{\mu})'}$$

Στόχος είναι η εκπαίδευση ενός 2Δ Gaussian Model $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ που να περιγράφει το δέρμα έχοντας ως δείγματα τις τιμές $I_{Cb}(x, y)$, $I_{Cr}(x, y)$ για το κάθε pixel των δειγμάτων δέρματος. Το \mathbf{c} στην παραπάνω σχέση είναι το διάνυσμα τιμών Cb & Cr $\mathbf{V}(x, y)$ σημείο της εικόνας. Επιπλέον, διευκρινίζεται ότι το διάνυσμα $\boldsymbol{\mu} = [\mu_{Cb}, \mu_{Cr}]^T$ αποτελεί την μέση τιμή και ότι ο πίνακας συνδιακύμανσης Σ προκύπτει από τα δείγματα δέρματος που δίνονται. Ακόμη, σημειώνεται ότι η κανονικοποίηση γίνεται στο $[0, 1]$.

Περιγραφή υλοποίησης:

Πρώτα, η συνάρτηση Gaussian εκπαιδεύει μια Gaussian Probability Density πάνω στα δείγματα με ανθρώπινο δέρμα που δίνονται στο αρχείο `skinSamplesRGB.mat`. Ειδικότερα, γίνεται μετατροπή από RGB σε YCbCr. Μετά, γίνεται συνένωση πινάκων ($n \times m \times 3$ σε RGB) σε ένα πίνακα 3 καναλιών (YCbCr) και στο τέλος αφαιρείται η Y συνιστώσα. Τελικά, η συνάρτηση επιστρέφει το διάνυσμα μέσης τιμής (mean) $\boldsymbol{\mu}$ (διάνυσμα 1×2) και τον πίνακα συνδιασποράς (covariance) Σ (πίνακας 2×2) της Gaussian Distribution.

Στο αρχείο `part1.py` φαίνονται αρχικά το διάβασμα των δειγμάτων και έπειτα η συνάρτηση Gaussian της οποίας τα αποτελέσματα φαίνονται παρακάτω:

Mean vector:

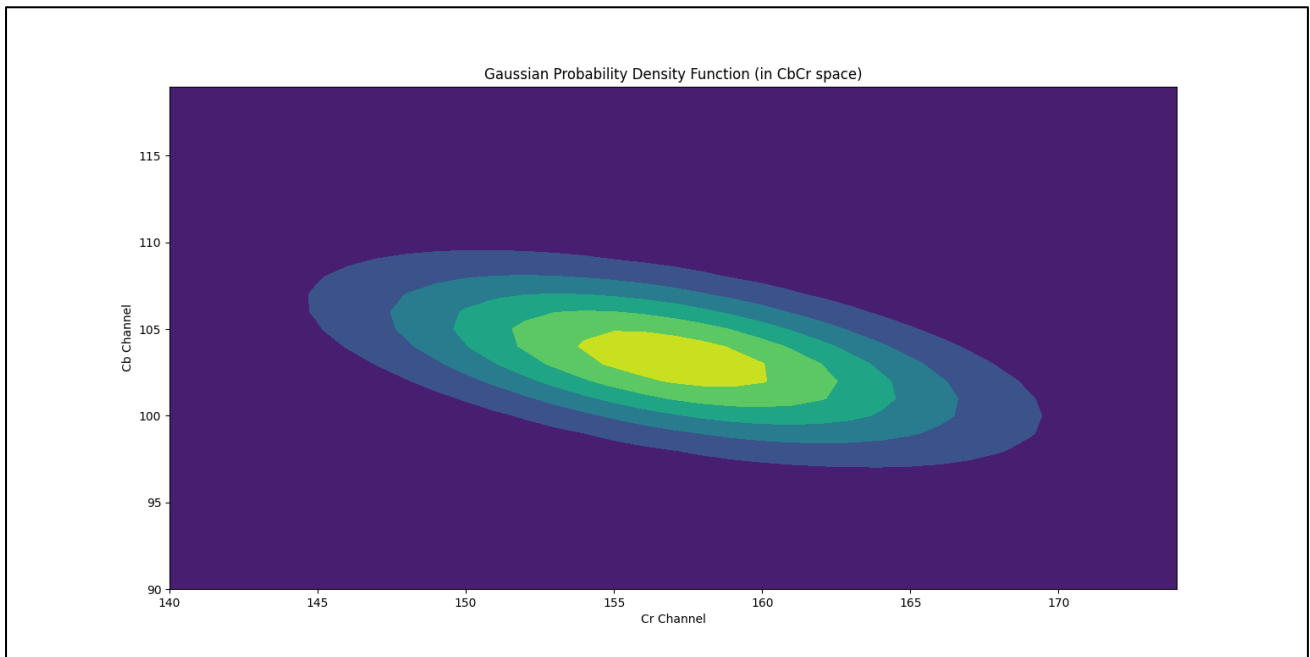
```
[157.04601571 103.2704826 ]
```

Covariance matrix:

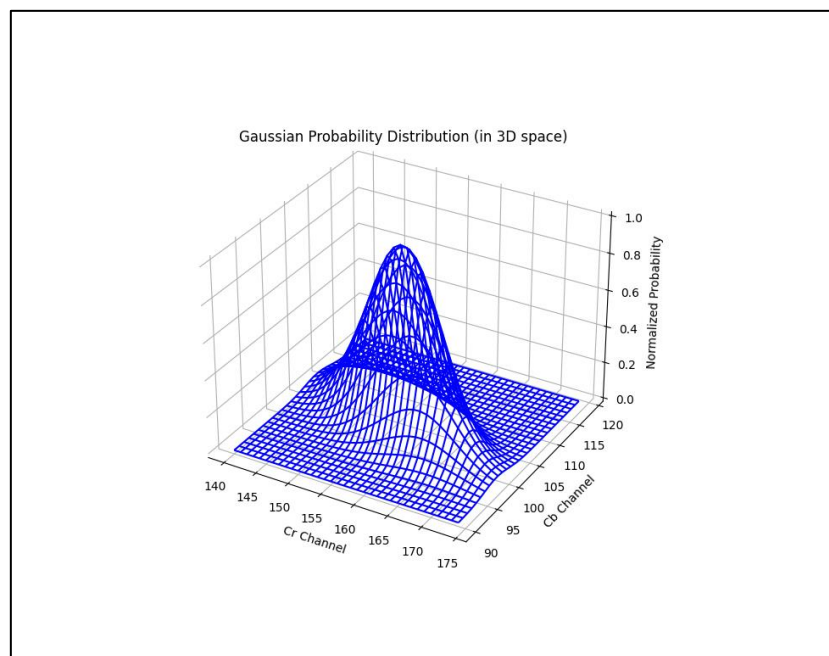
```
[[ 44.19103128 -11.9310385 ]
 [-11.9310385  11.19574811]]
```

Ακολούθως, φαίνεται η Gaussian Probability Density Function:

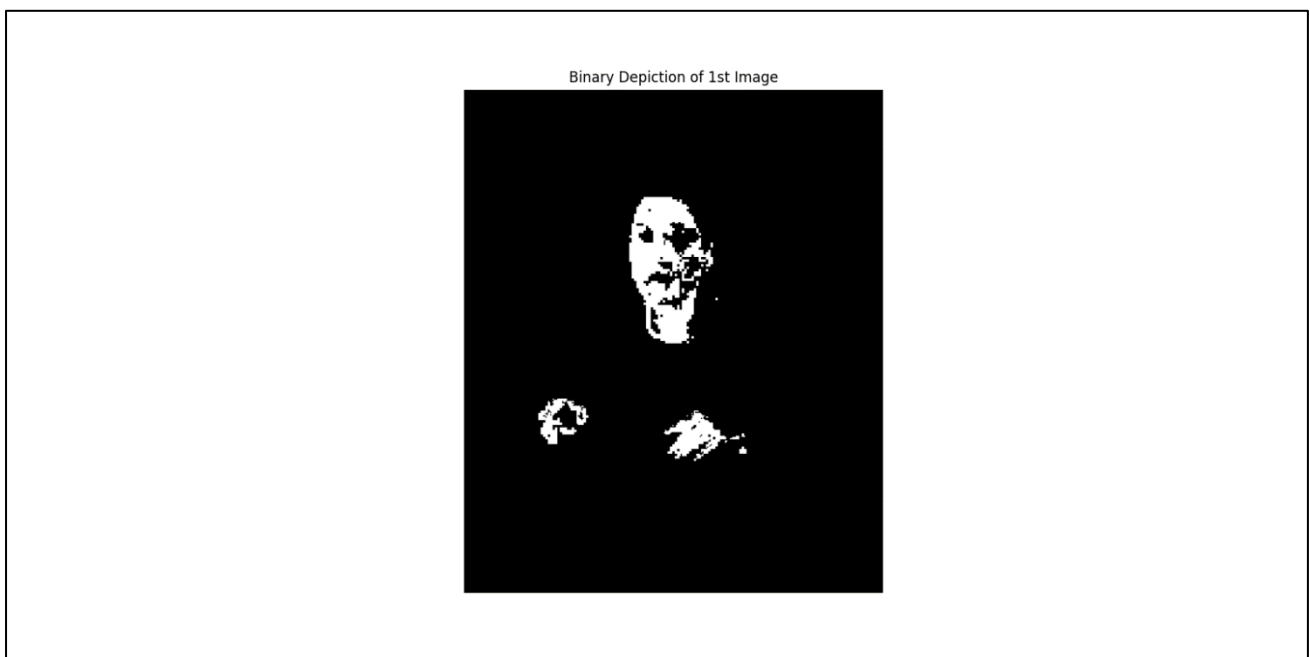
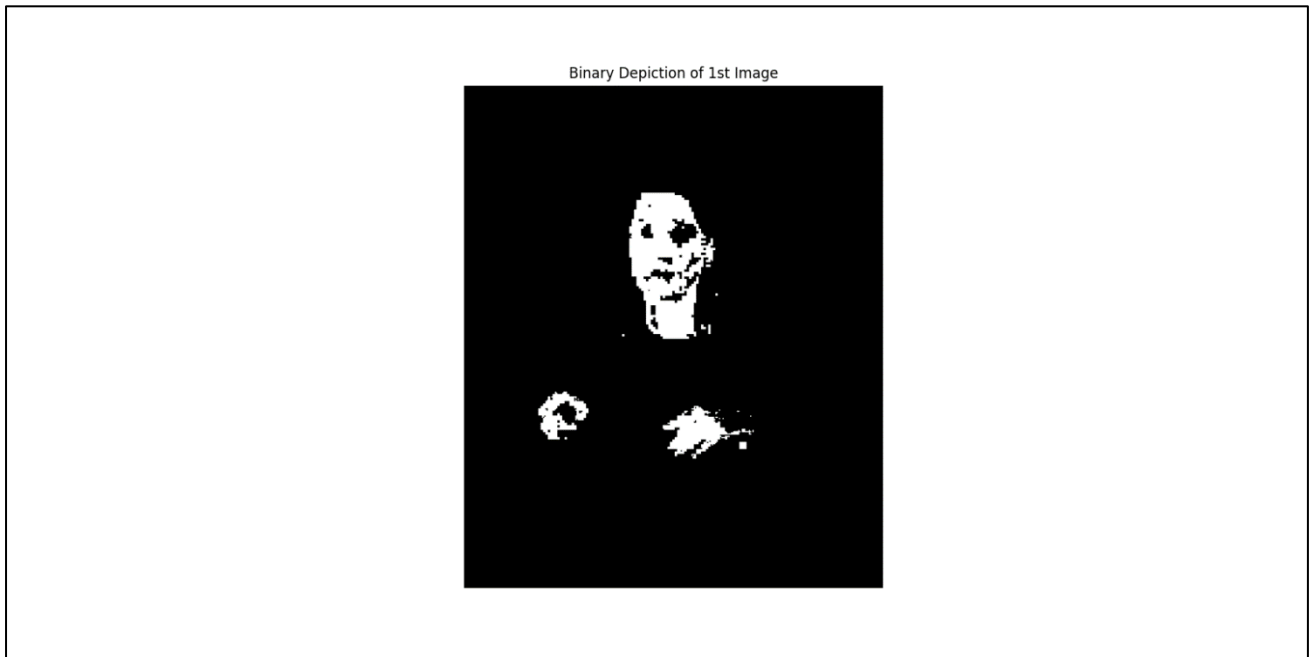
Στον Cb-Cr χώρο:



Στον 3Δ χώρο:

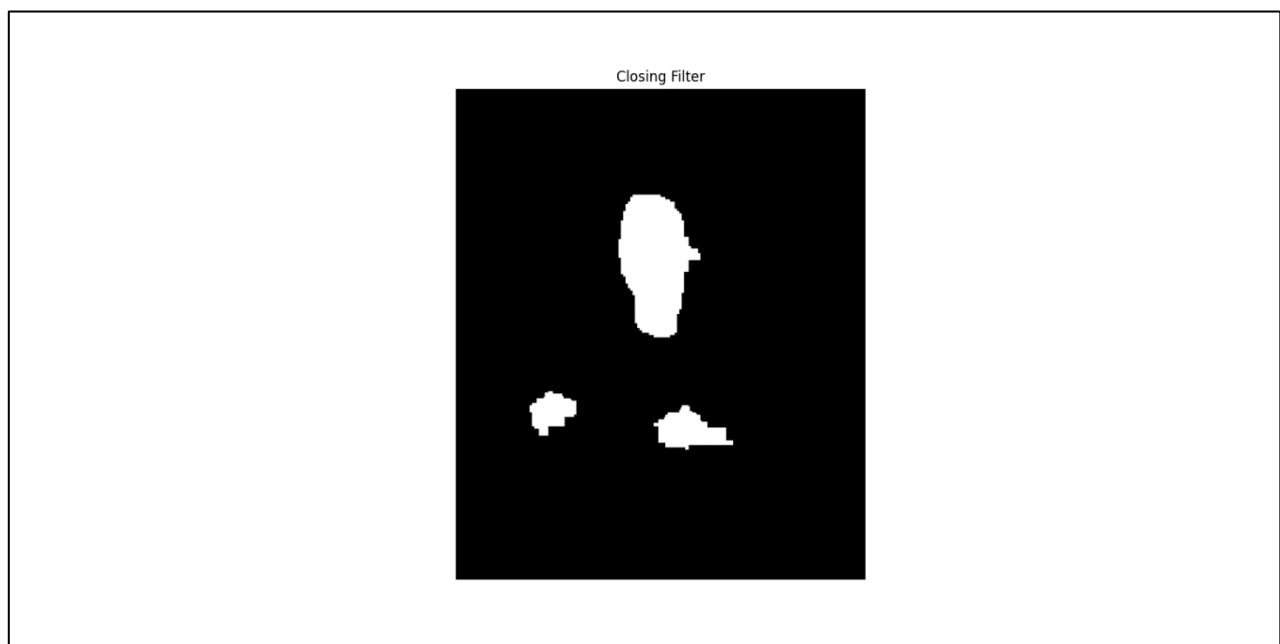
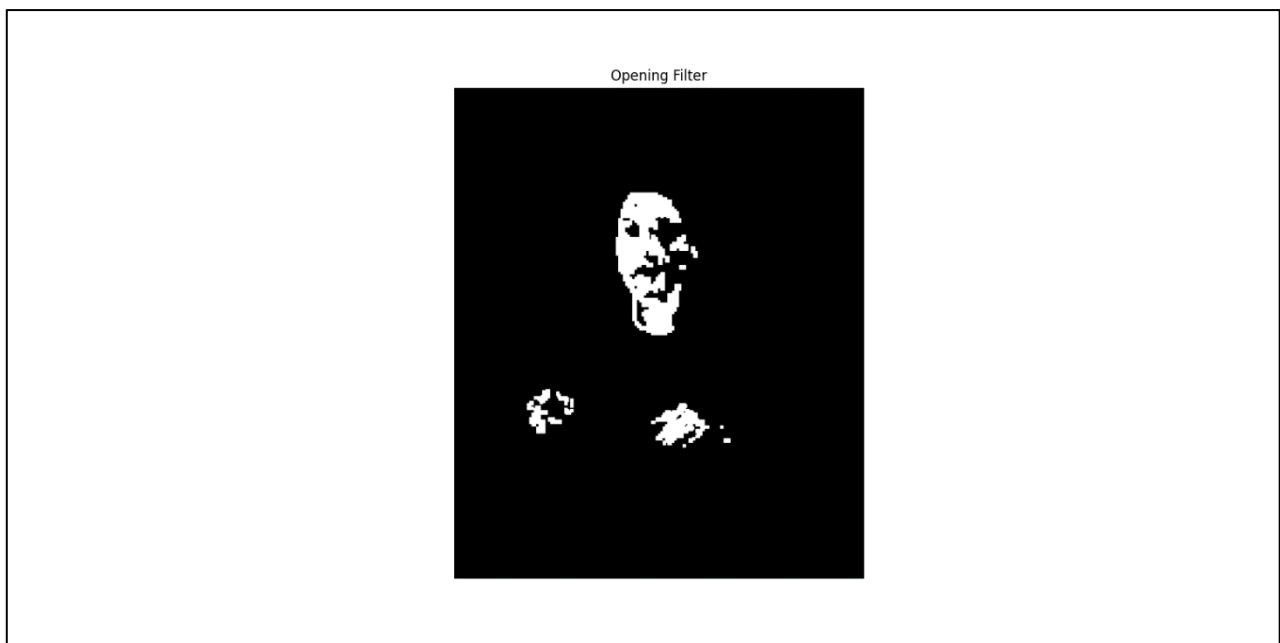


Έτσι, για την 1^η εικόνα, η δυαδική εικόνα του δέρματος προκύπτει με χρήση της $Pr(c(x, y) = \text{skin})$, $V(x, y)$ με κατωφλιοποίηση και με κανονικοποίηση της Gaussian Distribution με threshold της τάξης του 0.1 (1^η απεικόνιση) και 0.2 (2^η απεικόνιση). Η δυαδικές εικόνες που προκύπτουν, φαίνονται παρακάτω.

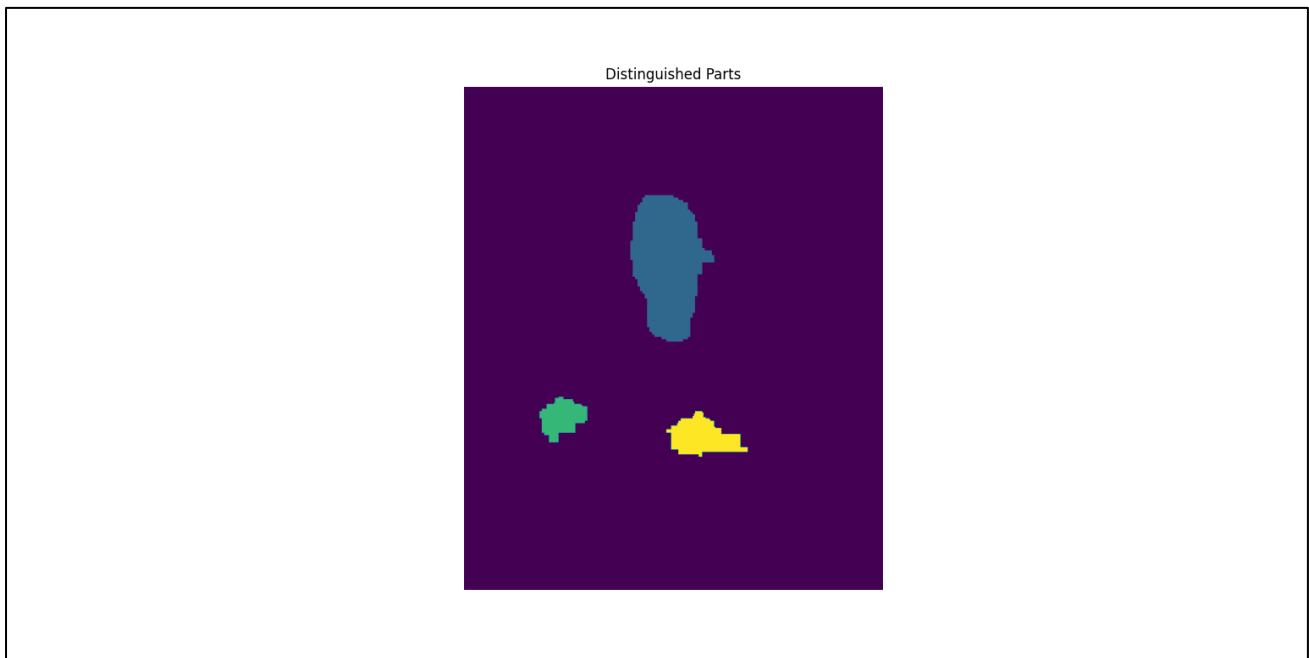


Αξίζει να σημειωθεί ότι οι περιοχές δέρματος X_1, \dots, X_k θα είναι δυαδικά σύνολα που είναι ξένα μεταξύ τους σε μια δυαδική εικόνα $X = \bigcup_{1 \leq i \leq k} X_i$. Από αυτές τις υποψήφιες περιοχές, στόχος είναι η ανίχνευση του προσώπου. Όμως, υπάρχουν και άλλα μέρη είτε με δέρμα είτε false positives τα οποία θα πρέπει να γίνει μορφολογική επεξεργασία της δυαδικής εικόνας ώστε να απομονωθεί το πρόσωπο. Για τον λόγο αυτό, εφαρμόζεται πρώτα opening με ένα πολύ μικρό δομικό στοιχείο B_o (3×3 pixels) για την απαλοιφή των πολύ μικρών περιοχών και έπειτα closing με ένα μεγάλο δομικό στοιχείο B_c (30×30 pixels) για το κλείσιμο των οπών όπου υπάρχουν. Οπότε, γίνεται διόρθωση μιας και καλύπτονται οι μικρές περιοχές και επιτυγχάνεται συνοχή και το αποτέλεσμα θα είναι της μορφής $Y = X \circ B_o \circ B_c = \bigcup_i Y_i$ & $Y_i \cap Y_j = \emptyset, \forall i \neq j$ και από αυτές τις περιοχές θα προκύψει το $F = \operatorname{argmax}_i \{\operatorname{card}(Y_i)\}$ και το bounding box $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ όπου εντοπίζεται το πρόσωπο.

Επομένως, εφαρμόζοντας στην δυαδική εικόνα opening (1^η απεικόνιση) και στη συνέχεια closing (2^η απεικόνιση) στο αποτέλεσμα.



Συνεπώς, τα βασικά σημεία της εικόνας, δηλαδή το πρόσωπο και το δεξί και αριστερό χέρι φαίνονται ξεκάθαρα και έτσι, μπορούν να εντοπιστούν με σχετική ευκολία.



Όπως μπορεί να δει κανείς, δημιουργούνται 3 σχήματα διαφορετικού χρώματος που σχετίζονται με τις περιοχές ενδιαφέροντος (τα λεγόμενα bounding boxes) τα οποία θα χρησιμοποιηθούν αργότερα για τον υπολογισμό της Οπτικής Ροής και για το Tracking του προσώπου και των χεριών.

Σχολιασμός:

Ως κατώφλι (threshold) ορίστηκε η τιμή $\tau = 0.2$. Η αύξησή του οδηγεί σε αποτελέσματα με λιγότερα true positives οπότε καλύτερη ακρίβεια ενώ η μείωσή του οδηγεί σε ανίχνευση περισσότερων false positives οπότε καλύτερο recall. Τα δομικά στοιχεία B_o & B_c που περιγράφονται και παραπάνω είναι τετράγωνα $n_o \times n_o$ & $n_c \times n_c$, και η αύξηση του n_o οδηγεί σε απόρριψη λιγότερων περιοχών ενώ η μείωση του n_c οδηγεί στο κλείσιμο μικρότερων σπών.

1.2 Παρακολούθηση Προσώπου και Χεριών

Η αρχικοποίηση των Bounding Boxes που περιλαμβάνει το πρόσωπο και τα χέρια της νοηματίστριας στη μορφή [x, y, width, height] είναι η εξής: Πρόσωπο: [138, 88, 73, 123], Αριστερό χέρι: [47, 243, 71, 66], Δεξί χέρι: [162, 264, 83, 48].

Τα Bounding Boxes είναι διευρυμένα, σχετικά με το αποτέλεσμα της ανίχνευσης του δέρματος, για να διευκολύνουν τη διαδικασία της παρακολούθησης. Οι παραπάνω διαστάσεις μπορούν να χρησιμοποιηθούν ανεξάρτητα από το αν έχει υλοποιηθεί το Μέρος 1.1 του εργαστηρίου.

Ζητείται να υλοποιηθούν οι αλγόριθμοι, όπως περιγράφονται παρακάτω, και να εφαρμοστούν για το πρόσωπο και για τα χέρια της νοηματίστριας. Το αν θα εφαρμοστούν για κάποιες/όλες τις περιοχές μαζί ή ξεχωριστά, αφέθηκε στην κρίση μας.

Αρχικά, πραγματοποιείται οριοθέτηση των πλαισίων της περιοχής ενδιαφέροντος. Τα x, y αποτελούν τις συντεταγμένες του πάνω αριστερά σημείου, width είναι το πλάτος και height είναι το ύψος του bounding box. Έτσι, προκύπτουν τα ακόλουθα αποτελέσματα:

Number of Distinguished Parts = 3 (Head, Right Hand, Left Hand)

Head Part: x = 139 , y = 91 , Width = 70 , Height = 122

Left Hand Part: x = 63 , y = 259 , Width = 40 , Height = 38

Right Hand Part: x = 169 , y = 271 , Width = 68 , Height = 38

Όμως, για την απλούστευση της διαδικασίας του monitoring της κίνησης, γίνεται χρήση των δοσμένων Bounding Boxes χωρίς όμως να υπάρχει μεγάλη απόκλιση.

Head Part: x = 138 , y = 88 , Width = 73 , Height = 123

Left Hand Part: x = 47 , y = 243 , Width = 71 , Height = 66

Right Hand Part: x = 162 , y = 264 , Width = 83 , Height = 48

1.2.1 Υλοποίηση του Αλγόριθμου των Lucas-Kanade

Σε μια ακολουθία εικόνων N frames $I_n(x)$, όπου $n = 1, \dots, N$ και $x = (x, y)$, το πεδίο οπτικής ροής $-d$, όπου $d(x) = (d_x, d_y)$, φέρνει σε αντιστοιχία δύο διαδοχικές εικόνες, έτσι ώστε:

$$I_n(\mathbf{x}) \approx I_{n-1}(\mathbf{x} + \mathbf{d}) \quad (2)$$

Ο αλγόριθμος των Lucas-Kanade υπολογίζει την οπτική ροή σε κάθε σημείο της εικόνας x με τη μέθοδο των ελάχιστων τετραγώνων, θεωρώντας ότι το d είναι σταθερό σε ένα μικρό παράθυρο γύρω από το σημείο και ελαχιστοποιώντας το τετραγωνικό σφάλμα:

$$J_{\mathbf{x}}(\mathbf{d}) = \int_{\mathbf{x}' \in \mathbb{R}^2} G_{\rho}(\mathbf{x} - \mathbf{x}') [I_n(\mathbf{x}') - I_{n-1}(\mathbf{x}' + \mathbf{d})]^2 d\mathbf{x}' \quad (3)$$

Όπου $G_{\rho}(x)$ είναι μια συνάρτηση παραθύρωσης, π.χ. Γκαουσιανή με τυπική απόκλιση ρ .

Έστω μια εκτίμηση d_i για το d και γίνεται προσπάθεια να βελτιωθεί κατά u , δηλαδή $d_{i+1} = d_i + u$. Αναπτύσσοντας κατά Taylor την έκφραση $I_{n-1}(x+d) = I_{n-1}(x+d_i+u)$ γύρω από το σημείο $x+d_i$, προκύπτει

$$I_{n-1}(\mathbf{x} + \mathbf{d}) \approx I_{n-1}(\mathbf{x} + \mathbf{d}_i) + \nabla I_{n-1}(\mathbf{x} + \mathbf{d}_i)^T \mathbf{u} \quad (4)$$

Βάζοντας αυτήν την έκφραση στην Εξ. (3) μπορεί ναδειχθεί ότι η λύση ελάχιστων τετραγώνων για τη βελτίωση της εκτίμησης της οπτικής ροής σε κάθε σημείο είναι:

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} (G_{\rho} * A_1^2)(\mathbf{x}) + \epsilon & (G_{\rho} * (A_1 A_2))(\mathbf{x}) \\ (G_{\rho} * (A_1 A_2))(\mathbf{x}) & (G_{\rho} * A_2^2)(\mathbf{x}) + \epsilon \end{bmatrix}^{-1} \cdot \begin{bmatrix} (G_{\rho} * (A_1 E))(\mathbf{x}) \\ (G_{\rho} * (A_2 E))(\mathbf{x}) \end{bmatrix} \quad (5)$$

$$A(\mathbf{x}) = [A_1(\mathbf{x}) \quad A_2(\mathbf{x})] = \left[\frac{\partial I_{n-1}(\mathbf{x} + \mathbf{d}_i)}{\partial x} \quad \frac{\partial I_{n-1}(\mathbf{x} + \mathbf{d}_i)}{\partial y} \right]$$

$$\text{όπου} \quad E(\mathbf{x}) = I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} + \mathbf{d}_i) \quad (6) \ \& \ (7)$$

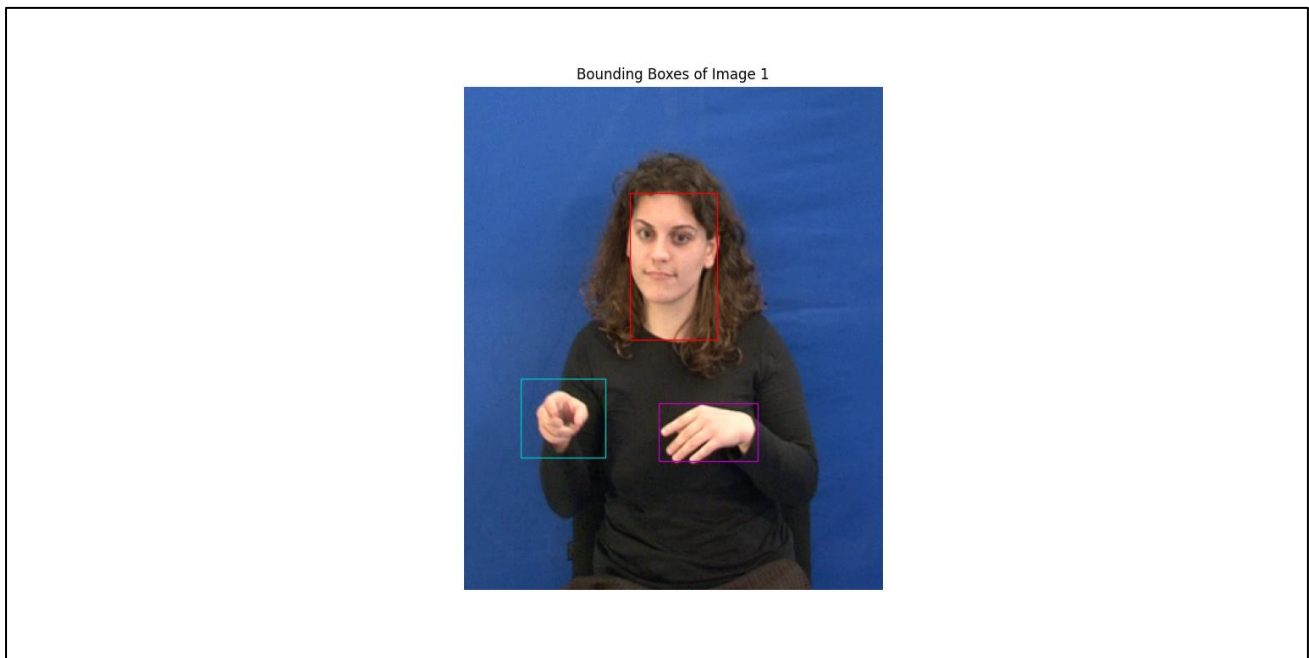
Και το $*$ δηλώνει συνέλιξη. Η μικρή θετική σταθερά ϵ βελτιώνει το αποτέλεσμα σε επίπεδες περιοχές με μειωμένη υφή και άρα μειωμένη πληροφορία για τον υπολογισμό της οπτικής ροής. Η ανανέωση του διανύσματος οπτικής ροής $d_{i+1} = d_i + u$, με το u να υπολογίζεται από την Εξ. (5), επαναλαμβάνεται αρκετές φορές ως τη σύγκλιση.

Ζητείται η υλοποίηση του αλγορίθμου των Lucas-Kanade σε Python. Ο αλγόριθμος υλοποιείται ως αυτόνομη συνάρτηση, που να δέχεται ως εισόδους δύο εικόνες (κομμένα παράθυρα με βάση το bounding box από δύο διαδοχικά πλαίσια του βίντεο), ένα σύνολο από σημεία ενδιαφέροντος (γωνίας) εντός του παραθύρου, το εύρος ρ του γκαουσιανού παραθύρου, τη θετική σταθερά κανονικοποίησης ϵ , και την αρχική εκτίμηση d_0 για το πεδίο οπτικής ροής, και επιστρέφει το d , π.χ.

$$[d_x, d_y] = lk(I1, I2, features, rho, epsilon, d_x0, d_y0)$$

➤ Βοήθεια για Python: (numpy) `meshgrid`, (cv2) `getGaussianKernel`, `filter2D`, (scipy.ndimage) `map_coordinates`, (matplotlib.pyplot) `quiver`.

Εφαρμόζοντας την παραπάνω οριοθέτηση, εντοπίζονται τα τρία ορθογώνια στην αρχική εικόνα που περιβάλλουν τις περιοχές ενδιαφέροντος (Bounding Boxes) και η εικόνα πλέον θα είναι:



Έπειτα, ακολουθεί η υλοποίηση του αλγορίθμου Monitoring Optical Flow και η εφαρμογή του στις δοθείσες εικόνες. Αυτή η διαδικασία επιτυγχάνεται με τον Lucas-Kanade Algorithm. Όπως αναφέρεται και παραπάνω, για τον υπολογισμό της συνάρτησης και των μερικών παραγώγων της εφαρμόζεται γραμμική παρεμβολή. Επιπλέον, η ανανέωση του διανύσματος $d_{i+1} = d_i + u$, επαναλαμβάνεται έως ότου υπάρξει σύγκλιση. Για ταχύτερη σύγκλιση, εφαρμόστηκε το Κριτήριο της Κατωφλιοποίησης για 2 διαδοχικές επαναλήψεις (αν $L2_{norm}(u) < 0.02$ τότε τερματίζει). Επιπροσθέτως, ο υπολογισμός της Optical Flow εστιάζει μόνο σε συγκεκριμένα σημεία ενδιαφέροντος της εικόνας I_2 χρησιμοποιώντας έναν Shi-Tomasi ανιχνευτή.

Για την ανίχνευση του προσώπου της νοσηματίστριας με χρήση του αλγορίθμου Lucas-Kanade για την εύρεση της οπτικής ροής ισχύει ότι I_n είναι ένα διανυσματικό πεδίο $d = (d_x(x, y), d_y(x, y))^T$ το οποίο υποδεικνύει την κίνηση των διαφόρων σημείων στο χώρο. Για μια ακολουθία εικόνων (super-image) $I(x, y, t)$ η εξίσωση της οπτικής ροής θα είναι: $I_x u + I_y v + I_t = 0$ όπου $I_x = \partial I / \partial x$, $I_y = \partial I / \partial y$, $I_t = \partial I / \partial t$ οι χωρο-χρονικές παράγωγοι της ακολουθίας εικόνων $I(x, y, t)$. Οι λύσεις της παραπάνω εξίσωσης ως προς u, v είναι άπειρες και σχηματίζουν οικογένεια ευθειών και για αυτό επιλέγεται ως λύση η οπτική ροή ανάμεσα σε δυο διαδοχικά καρέ $I_{n-1}(x, y)$, $I_n(x, y)$ ενός βίντεο. Οι Lucas-Kanade πρότειναν το τετραγωνικό κριτήριο κόστους (3) για την επίλυση του προβλήματος το οποίο λύνεται επαναληπτικά με την μέθοδο ελαχίστων τετραγώνων. Έτσι, για τα δυο πρώτα frames του βίντεο που δίνεται παρατηρείται ότι είναι εφικτή η διάκριση των κινήσεων των χεριών και του προσώπου στο διανυσματικό διάγραμμα και στην ενέργεια ενώ η φάση δεν δείχνει να έχει κάποια ουσιαστική πληροφορία.

Ύστερα από το normalization της εικόνας, από την εκτέλεση του αλγορίθμου Lucas-Kanade προκύπτουν οι παρακάτω απεικονίσεις για κάθε body part.

Image 1: Optical Flow of Head.

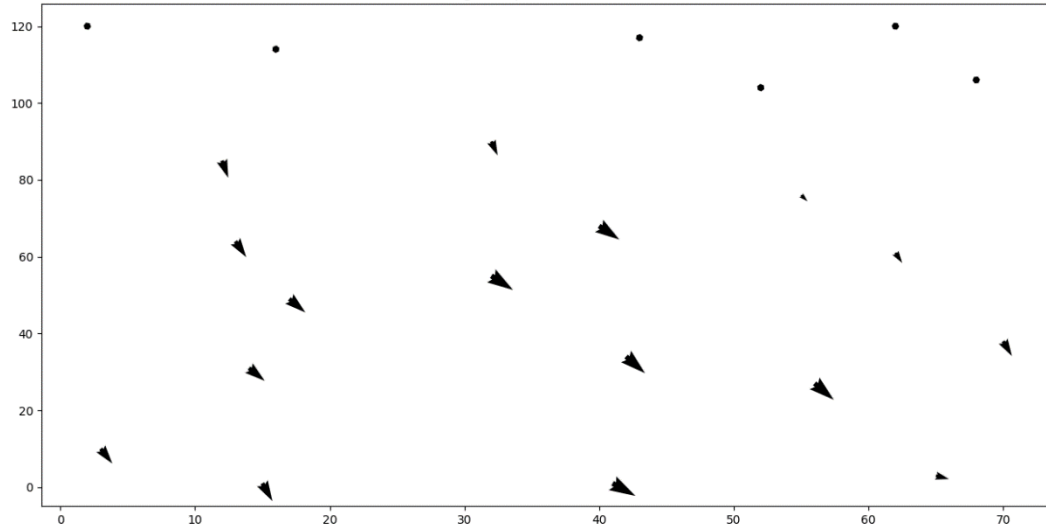
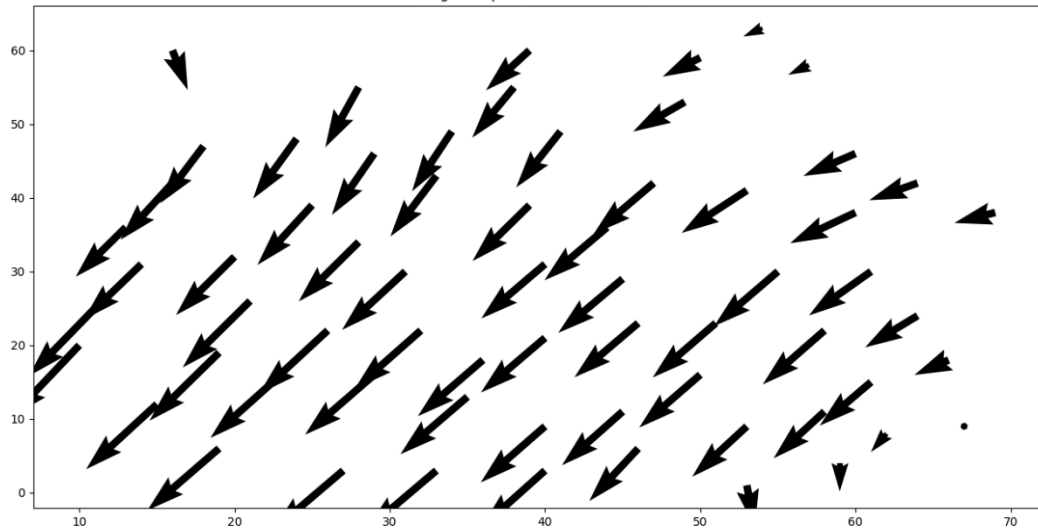
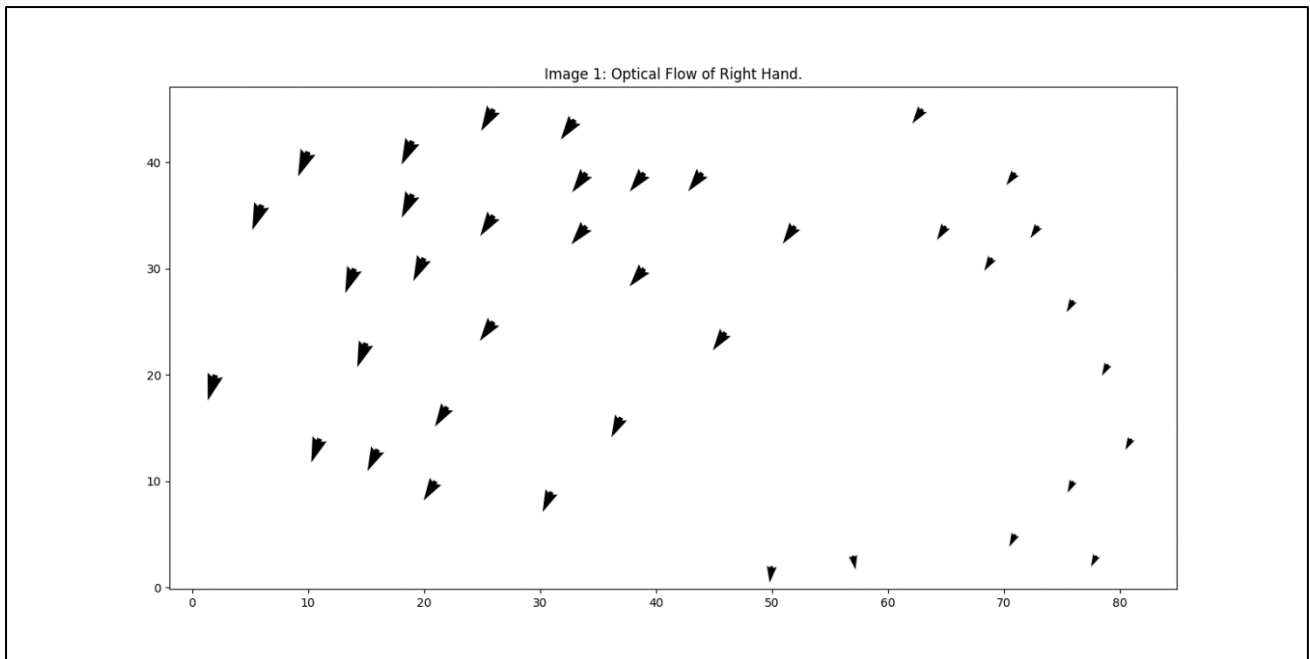


Image 1: Optical Flow of Left Hand.





Σχετικά με την υλοποίησή του, έχει δημιουργηθεί η συνάρτηση `lucas_kanade`. Αρχικά δημιουργείται το 2D grid, μετά υπολογίζονται οι μερικές παράγωγοι και έπειτα ορίζεται ο Gaussian πυρήνας για να γίνει η συνέλιξη. Έπειτα, για κάθε pixel στα features υπολογίζεται το interpolation, μετά η εξίσωση (7) και ορίζεται ο 1^{ος} και 2^{ος} πίνακας και αν υπάρχει σύγκλιση, επιστρέφει τους δυο πίνακες.

Σχολιασμός:

Αρχικά, παρατηρείται ότι η επικρατέστερη αρχική κίνηση είναι εκείνη του left hand.

Στην συνέχεια, επισημαίνεται ότι η παραγωγή με απλές διαφορές είναι μια διαδικασία που επηρεάζεται από τον θόρυβο. Οπότε, για να είναι ευσταθής, απαιτείται εξομάλυνση της εικόνας και αυτό πραγματοποιείται με την εφαρμογή smoothing πριν τον υπολογισμό της, γεγονός που ισοδυναμεί με convolution με παράγωγο Gaussian συνάρτησης (DoG).

Έπειτα, οι παράμετροι ϵ και ρ επηρεάζουν τον αλγόριθμο καθώς αφενός, το ϵ καθορίζει το smoothing των περιοχών στις οποίες $\|\nabla I\| \approx 0$ και αφετέρου, με την αύξηση του ρ , αυξάνεται το kernel size και το variance της Gaussian G_ρ που οδηγεί σε πιο εύρωστο tracking (όχι απαραίτητα λεπτομερές) ενώ η μείωση του ρ , οδηγεί σε πιο ευαίσθητο tracking εξαιτίας του θορύβου.

Τέλος, υπογραμμίζεται ότι ο αλγόριθμος συγκλίνει μετά από μερικές επαναλήψεις (~50) και αυτή πραγματοποιείται είτε με brute force επιλογή των επαναλήψεων είτε με βάση το Κριτήριο:

$$\max_{x,y \in D} \{u_x^2(x,y) + u_y^2(x,y)\} \leq \tau.$$

Πειραματισμός:

Σχολιασμός:

Αρχικά, καθώς το ρ αυξάνεται και το ϵ παραμένει σταθερό, το διανυσματικό πεδίο της Οπτικής Ροής εξομαλύνεται και αποκτά περισσότερες λεπτομέρειες. Αυτό συμβαίνει διότι, η αύξηση της τυπικής απόκλισης της Gaussian με την οποία γίνεται η συνέλιξη των παραγώγων των εικόνων οδηγεί σε μεγαλύτερη εξομάλυνση και συνεπώς, καλύτερη ανίχνευση κίνησης.

Από την άλλη, καθώς το ϵ μεταβάλλεται και το ρ παραμένει σταθερό, συμβαίνουν τα εξής. Σε επίπεδες περιοχές με μειωμένη υφή, το αποτέλεσμα βελτιώνεται. Όμως, υπερβολική αύξησή του οδηγεί σε μικρότερες μετατοπίσεις. Αυτό δικαιολογείται από το γεγονός ότι για μεγάλες τιμές του ϵ και για μικρές τιμές threshold του Κριτηρίου Σύγκλισης, ο αλγόριθμος καταλήγει να τερματίζει πριν την σύγκλιση οπότε επιστρέφει διανύσματα Οπτικής Ροής με χαμηλές τιμές.

1.2.2 Υπολογισμός της Μετατόπισης των Παραθύρων από τα Διανύσματα Οπτικής Ροής

Έχοντας υπολογίσει την οπτική ροή της εικόνας I_n στα σημεία που ορίζουν τα σημεία ενδιαφέροντος εντός του bounding box της εικόνας I_{n-1} , απομένει να βρεθεί το συνολικό διάνυσμα μετατόπισης του bounding box ορθογωνίου, με όσο το δυνατόν μεγαλύτερη ακρίβεια. Είναι γνωστό ότι τα διανύσματα οπτικής ροής έχουν κατά κανόνα μεγαλύτερο μήκος σε σημεία που ανήκουν σε περιοχές με έντονη πληροφορία υφής (π.χ. ακμές, κορυφές) και σχεδόν μηδενικό μήκος σε σημεία που ανήκουν σε περιοχές με ομοιόμορφη και επίπεδη υφή. Έτσι, καθώς η πλειονότητα των σημείων ενδιαφέροντος (γωνίες) βρίσκονται σε σημεία με έντονη υφή, είναι δυνατή η χρήση της μέσης τιμής των διανυσμάτων μετατόπισης. Όμως, για την επίτευξη καλύτερης ακρίβειας ή για την απόρριψη outliers, μπορούν να εφαρμοστούν εναλλακτικά κριτήρια, όπως για παράδειγμα η υλοποίηση της μέσης τιμής των διανυσμάτων μετατόπισης που έχουν ενέργεια μεγαλύτερη από μια τιμή κατωφλίου. Ως ενέργεια διανύσματος ταχύτητας ορίζεται $||d||^2 = d_x^2 + d_y^2$.

Ζητείται η υλοποίηση της συνάρτησης που θα δέχεται ως είσοδο τα διανύσματα της οπτικής ροής και θα υπολογίζει το τελικό διάνυσμα μετατόπισης του ορθογωνίου, π.χ.

$$[displ_x, displ_y] = displ(d_x, d_y)$$

Ζητείται η εκτέλεση του συνολικού συστήματος παρακολούθησης προσώπου και χεριών για την ακολουθία βίντεο που περιέχονται στο αρχείο `GreekSignLanguage.zip`. Πρόκειται για βίντεο ελληνικής νοηματικής γλώσσας γυρισμένο σε στούντιο με ελεγχόμενο φωτισμό. Αφήνεται στην κρίση μας ο πειραματισμός με εναλλακτικά κριτήρια υπολογισμού της μετατόπισης από τα διανύσματα οπτικής ροής.

Παρακάτω φαίνονται οι συνολικές μετατοπίσεις του bounding box που αντιστοιχεί στο left hand για τα πρώτα 2 frames χρησιμοποιώντας το παραπάνω Κριτήριο και διάφορες παραμετροποιήσεις.

Για μεγαλύτερη ακρίβεια στα αποτελέσματα και για εξάλειψη των outliers, εφαρμόζεται ως εναλλακτικό Κριτήριο της εξαγωγή της συνολικής μετατόπισης από την μέση τιμή των διανυσμάτων Οπτικής Ροής που έχουν ενέργεια μεγαλύτερη από μια τιμή κατωφλίου. Ως ενέργεια διανύσματος ταχύτητας ορίζεται: $||d||^2 = d_x^2 + d_y^2$.

Σε αυτή την περίπτωση, τα αποτελέσματα θα είναι:

```
epsilon = 0.01, rho = 1, Threshold = 0.001: dx = -1, dy = -2  
epsilon = 0.01, rho = 3, Threshold = 0.001: dx = -2, dy = -3  
epsilon = 0.01, rho = 5, Threshold = 0.001: dx = -2, dy = -3  
epsilon = 0.05, rho = 1, Threshold = 0.001: dx = -2, dy = -3  
epsilon = 0.05, rho = 3, Threshold = 0.001: dx = -2, dy = -3  
epsilon = 0.05, rho = 5, Threshold = 0.001: dx = -2, dy = -3
```

```
epsilon = 0.01, rho = 1, Threshold = 0.2: dx = -2, dy = -3  
epsilon = 0.01, rho = 3, Threshold = 0.2: dx = -2, dy = -3  
epsilon = 0.01, rho = 5, Threshold = 0.2: dx = -2, dy = -3  
epsilon = 0.05, rho = 1, Threshold = 0.2: dx = -2, dy = -3  
epsilon = 0.05, rho = 3, Threshold = 0.2: dx = -2, dy = -3  
epsilon = 0.05, rho = 5, Threshold = 0.2: dx = -2, dy = -3
```

```
epsilon = 0.01, rho = 1, Threshold = 0.5: dx = -2, dy = -4  
epsilon = 0.01, rho = 3, Threshold = 0.5: dx = -2, dy = -4  
epsilon = 0.01, rho = 5, Threshold = 0.5: dx = -2, dy = -4  
epsilon = 0.05, rho = 1, Threshold = 0.5: dx = -2, dy = -4  
epsilon = 0.05, rho = 3, Threshold = 0.5: dx = -2, dy = -4  
epsilon = 0.05, rho = 5, Threshold = 0.5: dx = -2, dy = -4
```

```
epsilon = 0.01, rho = 1, Threshold = 0.7: dx = -2, dy = -4  
epsilon = 0.01, rho = 3, Threshold = 0.7: dx = -2, dy = -4  
epsilon = 0.01, rho = 5, Threshold = 0.7: dx = -2, dy = -4  
epsilon = 0.05, rho = 1, Threshold = 0.7: dx = -2, dy = -4  
epsilon = 0.05, rho = 3, Threshold = 0.7: dx = -2, dy = -4  
epsilon = 0.05, rho = 5, Threshold = 0.7: dx = -2, dy = -4
```


Σχολιασμός:

Παρατηρείται ότι όταν το κατώφλι ενέργειας παίρνει μεγάλες τιμές, τότε η συνολική μετατόπιση εξάγεται ως η μέση τιμή διανυσμάτων Οπτικής Ροής με αρκετά αυξημένη ενέργεια. Συνεπώς, η εξαγόμενη μετατόπιση των bounding boxes είναι μεγαλύτερη από την πραγματική. Επιπλέον, η ανίχνευση κίνησης είναι πολύ πιθανό να μην είναι εφικτή εξαιτίας αυτών των μετατοπίσεων.

Από την άλλη, όταν το κατώφλι ενέργειας παίρνει μικρές τιμές, τότε το Κριτήριο Εξαγωγής Συνολικών Μετατοπίσεων εκφυλλίζεται στην περίπτωση που λαμβάνεται υπόψιν η μέση τιμή όλων των διανυσμάτων Οπτικής Ροής. Συνεπώς, η ακρίβεια των αποτελεσμάτων είναι μειωμένη εξαιτίας της ύπαρξης σημείων που βρίσκονται σε περιοχές με ομοιόμορφη και επίπεδη υφή και δυσχεραίνεται ακόμη περισσότερο για μικρές τιμές των ϵ και ρ .

Όσο μικρότερο είναι το ϵ , τόσο πιο αραιή είναι η κίνηση στις διάφορες περιοχές, αφού φαίνονται όλο και λιγότερα διανύσματα μετατόπισης. Όμως, όσο μεγαλώνει το ϵ , επιτυγχάνεται το αντίθετο από πριν και επιπλέον, τα bounding boxes μετακινούνται ταχύτερα.

Όσο μικρότερο είναι το ρ , τόσο πιο απότομες είναι οι μεταβολές στη διεύθυνση των διανυσμάτων μετατόπισης που ανήκουν σε διαδοχικά pixels. Όμως, όσο μεγαλύτερο είναι το ρ , τόσο μεγαλώνει το μήκος των διανυσμάτων μετατόπισης του πεδίου οπτικής ροής και τόσο καλύτερος είναι ο εντοπισμός του προσώπου.

Επιπλέον, σημειώνεται ότι υπάρχει ομαλότητα στο πεδίο οπτικής ροής που οφείλεται στην χρήση διαδοχικών καρέ της ακολουθίας βίντεο. Επίσης, ο εντοπισμός του κεφαλιού από το bounding box είναι επιτυχής ενώ των χεριών λιγότερο εξαιτίας πιθανών απότομων μεταβολών της κίνησής τους.

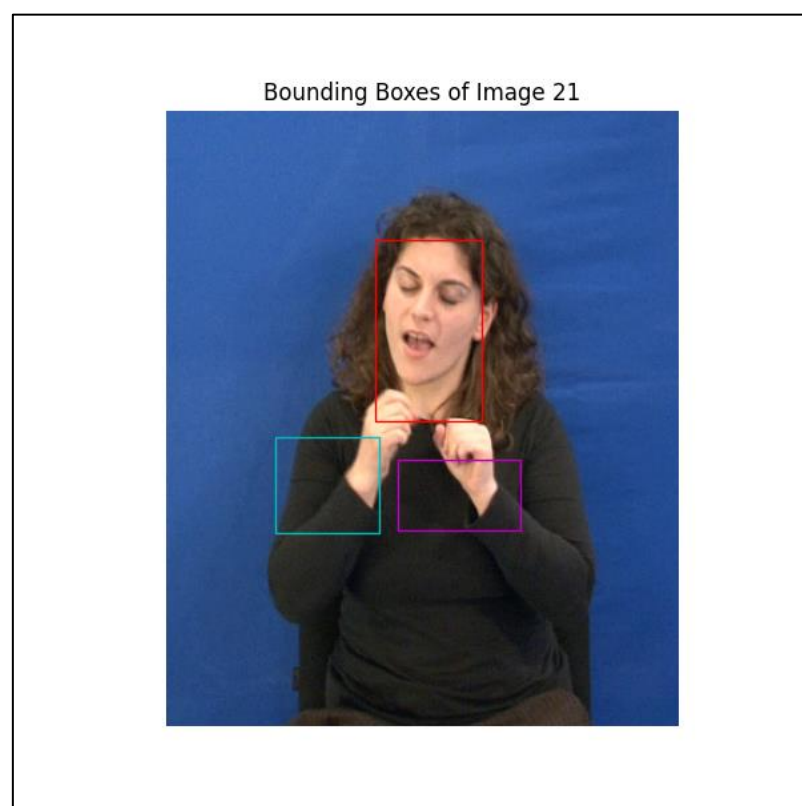
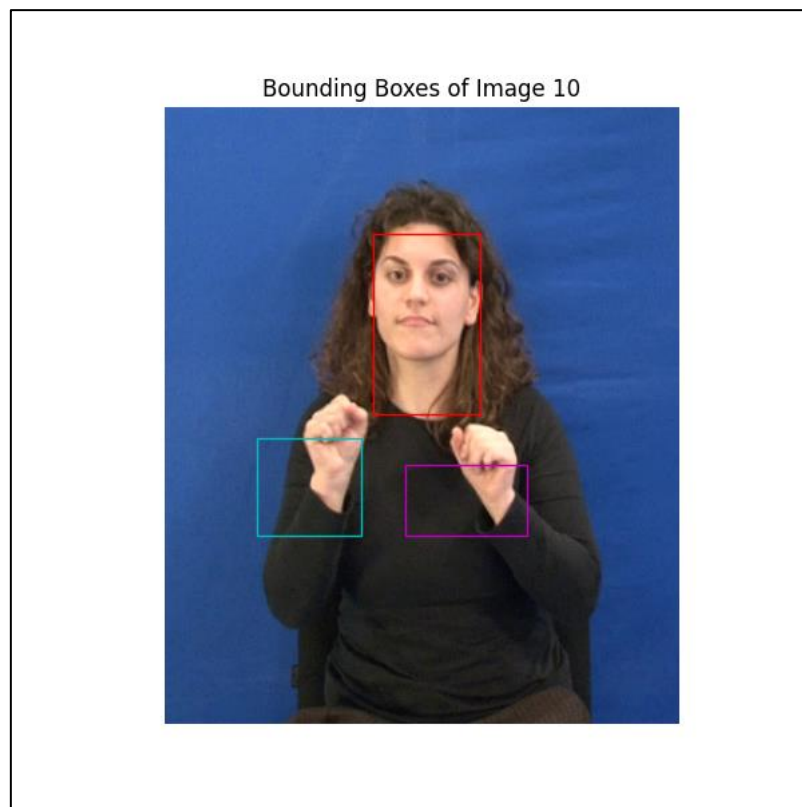
Εν κατακλείδι, μια ικανοποιητική περιοχή τιμών φαίνεται να είναι το διάστημα $[0.2, 0.5]$ με κατάλληλες, βέβαια, τιμές για ϵ και ρ .

Παρακάτω, απεικονίζονται κάποια στιγμιότυπα της ανίχνευσης της κίνησης για $\epsilon = 0.01$, $\rho = 8$ και Threshold = 0.5.

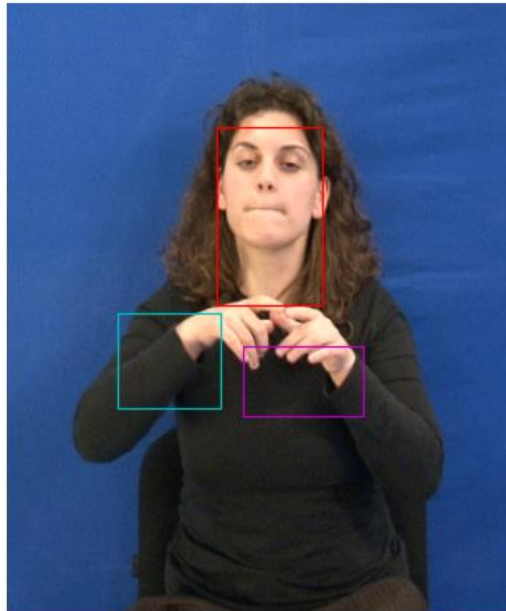
(Σημειώνεται ότι απαιτείται η δημιουργία ενός κενού φακέλου με όνομα “motion_tracking”).

Η συνάρτηση motion_tracking που φαίνεται στο Παράρτημα της αναφοράς, λειτουργεί ως εξής. Αρχικά, κάνει crop στα bounding boxes. Μετά, για κάθε frame, εξάγονται τα χαρακτηριστικά με τον Shi-Tomashi Detector, έπειτα εφαρμόζεται ο Lucas-Kanade αλγόριθμος, υπολογίζεται το speed Vector για κάθε bounding box, ανανεώνονται οι θέσεις τους και συνεχίζεται η διαδικασία αυτή μέχρι να τελειώσουν τα frames.

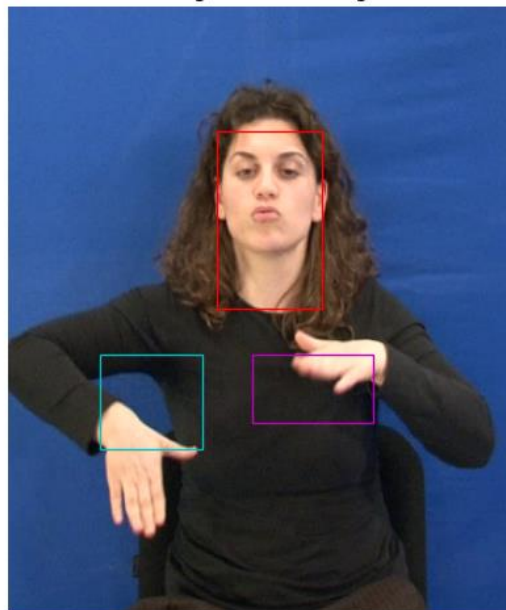
Μερικά από τα αποτελέσματα που προέκυψαν είναι τα ακόλουθα:



Bounding Boxes of Image 45



Bounding Boxes of Image 59



1.2.3 Πολύ-κλιμακωτός Υπολογισμός Οπτικής Ροής

Ζητείται η υλοποίηση της πολύ-κλιμακωτής εκδοχής του αλγορίθμου των Lucas-Kanade. Ο αλγόριθμος θα αναλύει τις αρχικές εικόνες σε γκαουσιανές πυραμίδες και θα υπολογίζει την οπτική ροή από τις πιο μικρές (τραχείς) στις πιο μεγάλες (λεπτομερείς) κλίμακες, χρησιμοποιώντας τη λύση της μικρής κλίμακας ως αρχική συνθήκη για τη μεγάλη κλίμακα. Η υλοποίηση του αλγορίθμου θα γίνει ως αυτόνομη συνάρτηση, παρόμοια με πριν, αλλά θα δέχεται επίσης ως είσοδο τον αριθμό των κλιμάκων της πυραμίδας, και θα χρησιμοποιεί τον αλγόριθμο των Lucas-Kanade μονής κλίμακας ως υπο-ρουτίνα.

Επιπλέον, ζητείται το τρέξιμο του πολύ-κλιμακωτού αλγορίθμου στην ίδια ακολουθία εικόνων και ο σχολιασμός των διαφορών που παρατηρούνται στην ταχύτητα σύγκλισης και στην ποιότητα του αποτελέσματος σε σχέση με τον αλγόριθμο μονής κλίμακας. Το ολικό displacement d για το bounding box υπολογίζεται όμοια με το 1.2.2.

Για τον υπολογισμό της Οπτικής Ροής σε μεγαλύτερες κινήσεις που υπερβαίνουν κατά πολύ το ένα pixel, δεν ισχύουν οι παραδοχές για τους όρους 1^{ης} τάξης του μονοκλιμακωτού αλγορίθμου Lucas-Kanade. Επομένως, ο αλγόριθμος των Lucas-Kanade επεκτείνεται και σε πολύ-κλιμακωτή εκδοχή N σταδίων, κατά την οποία αναλύονται οι αρχικές διαδοχικές εικόνες I_1 και I_2 σε Gaussian Πυραμίδες και στη συνέχεια υπολογίζεται το πεδίο Οπτικής Ροής από τις μεγαλύτερες στις μικρότερες κλίμακες. Αρχικά, επιλέγεται μια εικόνα μεγέθους $1/2^N$ της αρχικής, έπειτα εφαρμόζεται ο αλγόριθμος, μετά υποδειματοποιείται η οπτική ροή και τέλος, τίθεται ως input στο επόμενο στάδιο. Επιπλέον, για αυτή την μετάβαση εφαρμόζεται φιλτράρισμα με ένα Lowpass Gaussian Filter πριν την υποδειματοληψία για την αντιμετώπιση του aliasing. Αυτή η μέθοδος οδηγεί στην γρηγορότερη εύρεση πεδίων οπτικής ροής με καλή ποιότητα και ακρίβεια.

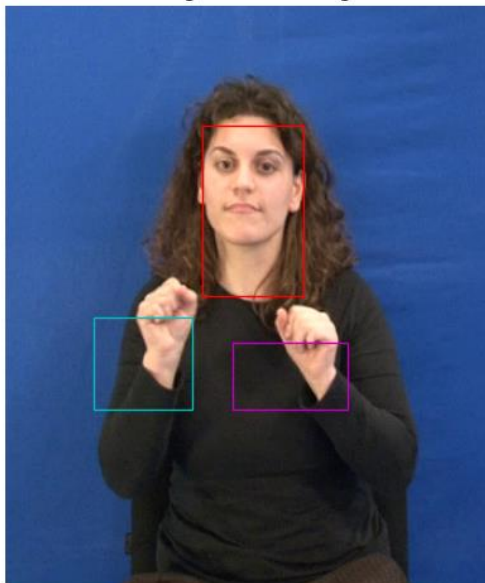
(Σημειώνεται ότι απαιτείται η δημιουργία ενός κενού φακέλου με όνομα “multi_tracking”).

Η συνάρτηση multi_lk λειτουργεί ως εξής. Αρχικά, αν το scale είναι μηδέν, τότε υπολογίζει τον μονοδιάστατο Lucas-Kanade ενώ διαφορετικά, υπολογίζει τον Πολυδιάστατο. Ειδικότερα, χρησιμοποιεί ένα Gaussian πυρήνα τριών pixel. Έστερα, για την αποφυγή του aliasing, χρησιμοποιεί ένα lowpass φίλτρο και εφαρμόζει υποδειματοληψία. Τέλος, γίνεται αφαίρεση των χαρακτηριστικών για χαμηλότερη ανάλυση.

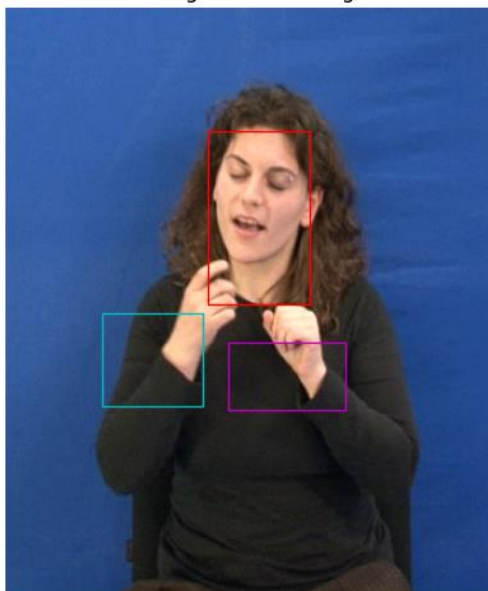
Ο σχετικός κώδικας για την παραγωγή όλων των frames βρίσκεται στο Παράρτημα αυτής της Αναφοράς.

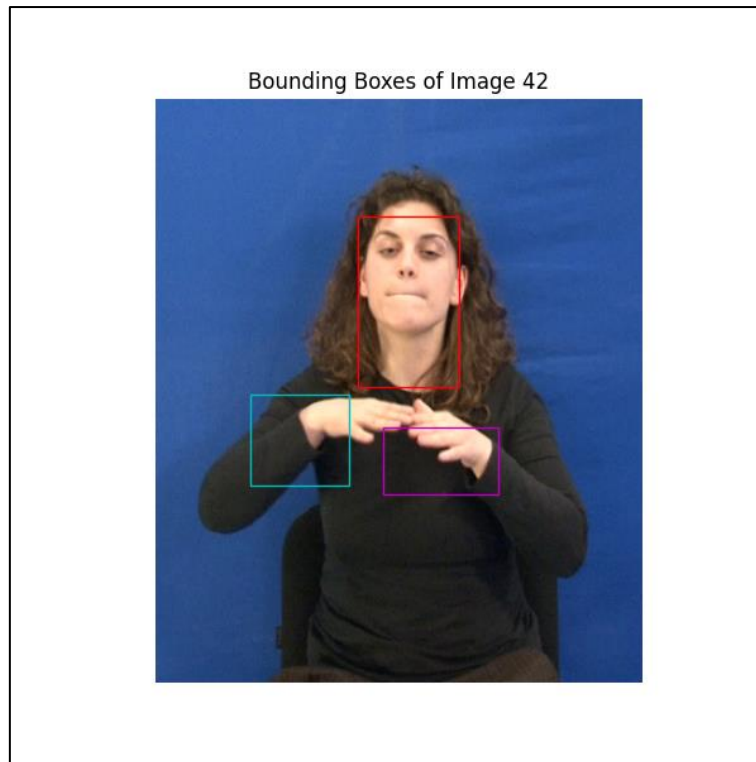
Ενδεικτικά, φαίνονται παρακάτω οι εικόνες που προέκυψαν.

Bounding Boxes of Image 10



Bounding Boxes of Image 22





Πειραματισμός:

Σχολιασμός:

Παρατηρείται ότι η πολυκλιμακωτή εκδοχή φαίνεται να ανιχνεύει και τις μεγάλες αλλά και τις μικρές κινήσεις.

Ακόμη, με κατάλληλη κλίμακα, παρατηρείται ότι οι επαναλήψεις που χρειάζονται για την Σύγκλιση της μεθόδου Lucas-Kanade λιγοστεύουν εφόσον η προηγούμενη κλίμακα παρέχει μια αρχική εκτίμηση που είναι αντιπροσωπευτική της κίνησης μεταξύ 2 frames. Όμως, για μεγάλο αριθμό κλιμάκων υπάρχει περίπτωση η συνολική μετατόπιση των bounding boxes να είναι ιδιαίτερα αυξημένη οπότε τότε χάνεται το tracking της κίνησης. Από τους πειραματισμούς προκύπτει ότι μια καλή επιλογή κλίμακας για την Gaussian Πυραμίδα είναι οι 3-4 κλίμακες.

Όσον αφορά το N , με την αύξησή του, επιτυγχάνεται ταχύτερη σύγκλιση των Lucas-Kanade διότι οι αρχικές συνθήκες σε κάθε στάδιο δίνουν συνεχώς καλύτερες αρχικές συνθήκες στον υπολογισμό του d .

Όσο μεγαλύτερη είναι η κλίμακα, τόσο καλύτερος είναι ο εντοπισμός των κινήσεων και τόσο καλύτερη είναι η ταχύτητα ως προς τη σύγκλιση και άρα την βελτίωση του αποτελέσματος.

Παρατηρείται, γενικότερα, ότι τα μορφολογικά φίλτρα έκαναν την κάλυψη οπών και την εξάλειψη μικρών περιοχών επιτυχώς. Τα αποτελέσματα ποικίλουν ανάλογα με τις παραμέτρους κατωφλίου αφού αν αυτή επιλεγεί να είναι μικρή, θα επιλεγούν επιπλέον περιοχές οι οποίες τελικά μπορεί να επηρεάσουν την τελική ανίχνευση αφού μπορεί να επιλεγεί κάποιο πλαίσιο το οποίο εκτός από το χέρι της νοηματίστριας, να περιέχει και κάτι άλλο που δεν είναι επιθυμητό, ενώ, αν αυτή είναι αρκετά μεγάλη, μπορεί να μην επιλεγεί η περιοχή του χεριού και τελικά να γίνει λάθος ανίχνευση. Επομένως, θα πρέπει να είναι προσεκτική η τιμή που θα επιλεγεί.

Μέρος 2: Εντοπισμός Χωρο-χρονικών Σημείων Ενδιαφέροντος και Εξαγωγή Χαρακτηριστικών σε Βίντεο Ανθρωπίνων Δράσεων

Αυτό το μέρος της εργαστηριακής άσκησης πραγματεύεται την εξαγωγή χωρο-χρονικών χαρακτηριστικών με στόχο την εφαρμογή τους στο πρόβλημα κατηγοριοποίησης βίντεο που περιέχουν ανθρώπινες δράσεις. Στα πλαίσια της άσκησης, δίνονται βίντεο από 3 κλάσεις δράσεων (walking, running, boxing) από τα οποία θα εξαχθούν χωρο-χρονικοί περιγραφητές με σκοπό την κατηγοριοποίηση των δράσεων που απεικονίζουν. Τα βίντεο διαβάζονται καλώντας την συνάρτηση `read_video(name, nframes, 0)` από το συμπληρωματικό υλικό, όπου `name` το πλήρες όνομα του βίντεο και `nframes` ο αριθμός των `frames` (π.χ. 200) που είναι θεμιτό να διαβαστούν. Τα `video` θα αναπαρίστανται με ένα τρισδιάστατο πίνακα, του οποίου η 3^η διάσταση αντιστοιχεί στον χρόνο και αποτελεί την ακολουθία των `frames`, τα οποία είναι grayscale εικόνες.

2.1 Χωρο-χρονικά Σημεία Ενδιαφέροντος

Οι ανιχνευτές τοπικών χαρακτηριστικών αναζητούν χωρο-χρονικά σημεία και κλίμακες ενδιαφέροντος, τα οποία αντιστοιχούν σε περιοχές που χαρακτηρίζονται από σύνθετη κίνηση ή απότομες μεταβολές στην εμφάνιση του `video` εισόδου. Αυτό επιτυγχάνεται μεγιστοποιώντας μια συνάρτηση 'οπτικής σημαντικότητας'. Πολλοί ανιχνευτές έχουν επινοηθεί τα τελευταία χρόνια αντλώντας αραιά αλλά εύρωστα σημεία. Το στάδιο αυτό της εργαστηριακής άσκησης, ασχολείται με 2 διαφορετικούς τέτοιους ανιχνευτές: 1) *Harris detector* & 2) *Gabor detector*.

2.1.1 Ζητείται ο υπολογισμός του ανιχνευτή *Harris* ο οποίος αποτελεί μια επέκταση σε 3 διαστάσεις του ανιχνευτή γωνιών *Harris-Stephens*, που υλοποιήθηκε στην 1^η Εργαστηριακή Άσκηση. Για κάθε `voxel` του βίντεο ζητείται ο υπολογισμός του 3×3 πίνακα $M(x, y, t)$ προσθέτοντας στον 2Δ δομικό τανυστή και τη χρονική παράγωγο:

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

ή σε μορφή πινάκων:

$$M(x, y, t; \sigma, \tau) = g(x, y, t; \sigma, \tau) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$$

όπου $g(x, y, t; \sigma, \tau)$ ένας 3Δ γκαουσιανός πυρήνας ομαλοποίησης και 3Δ γκαουσιανός πυρήνας ομαλοποίησης και $\nabla L(x, y, t; \sigma, \tau)$ οι χωρο-χρονικές παράγωγοι για την χωρική κλίμακα σ και τη χρονική κλίμακα τ . Οι παράγωγοι (χωρικές και χρονικές) μπορούν να υπολογιστούν εφαρμόζοντας συνέλιξη με τον πυρήνα κεντρικών διαφορών $[-1 \ 0 \ 1]^T$ (προσαρμοσμένο στην κατάλληλη διάσταση).

Το 3Δ κριτήριο γωνιότητας ακολουθεί και αυτό την ίδια λογική:

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

Σε αυτό το Μέρος, γίνεται επέκταση του ανιχνευτή Harris Stephens σε χώρο και χρόνο. Έστω μια ακολουθία εικόνων $I_t(x, y)$ στην οποία θα γίνεται αναφορά ως $I(x, t, t)$. Ο ανιχνευτής Harris Stephens ακολουθεί την ίδια λογική με τον κλασσικό ανιχνευτή Harris Stephens για γωνίες. Αρχικά, ορίζεται

το διάνυσμα παραγώγου: $g(x, y, t) = \begin{pmatrix} L_x \\ L_y \\ L_t \end{pmatrix} = \begin{pmatrix} I_\sigma *_{x} g_x \\ I_\sigma *_{y} g_y \\ I_\sigma *_{t} g_t \end{pmatrix}$ όπου g_x, g_y, g_t οι πυρήνες των κεντρικών

διαφορών της μορφής $g = (-1, 0, 1)^T$ και $*_x, *_y, *_t$ ο τελεστής της συνέλιξης κατά διάσταση. Για παράδειγμα, $I_\sigma(x, y, t) *_{t} g_t = \int_{t' \in \mathbb{R}} I_\sigma(x, y, t - t') g(x, y, t') dt'$.

Οι πυρήνες κεντρικών διαφορών, σε αντίθεση με την απλή διαφορά, υπερτερούν διότι προσεγγίζουν την παράγωγο κάνοντας χρήση δύο ακραίων τιμών. Αφού λοιπόν, δημιουργηθούν οι παράγωγοι L_x, L_y, L_t , προκύπτει ο πίνακας $M(x, y, t) = G(x, y, t | \sigma, \tau) * (g(x, y, t) g^T(x, y, t))$ για τον οποίο ορίζεται το Κριτήριο Σημαντικότητας: $H(x, y, t) = \det M(x, y, t) - k \text{tr}^3 M(x, y, t) = \lambda_1, \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3$.

Τα σημεία ενδιαφέροντος για το Harris Stephens ανιχνευτή θα ληφθούν ως τα πρώτα k ακρότατα του Κριτηρίου Σημαντικότητας. Αναλυτικότερα, το Κριτήριο H κανονικοποιείται $\hat{H}(x, y, t) = \frac{H(x, y, t) - H_{\min}}{H_{\max} - H_{\min}}$ ώστε να βρεθούν τα σημεία όπου $\hat{H}(x, y, t) \leq \tau$, όπου $\tau \in [0, 1]$. Ύστερα, διατηρούνται τα k πρώτα σημεία (ως προς τις τιμές) και τελικά προκύπτουν οι περιοχές ενδιαφέροντος ως τούπλες της μορφής: $(x, y, t, \sigma, \tau, \hat{H}(x, y, t))$.

Πρωταρχικά, ορίζονται οι τυπικές παράμετροι και γίνεται το διάβασμα των βίντεο.

Έπειτα, υλοποιούνται οι συναρτήσεις `Gaussian_1D` & `Gaussian_2D` που θα χρησιμεύσουν για τους Ανιχνευτές.

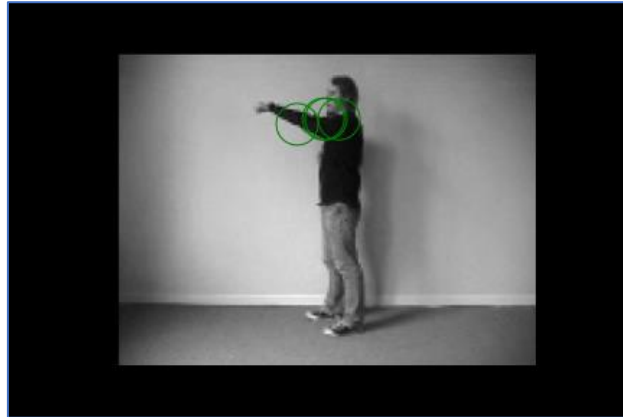
Στην συνέχεια, η συνάρτηση `HarrisDetector` υλοποιεί την εύρεση σημείων ενδιαφέροντος και επιστρέφει το Κριτήριο Σημαντικότητας για το βίντεο που δίνεται ως `input`. Πιο συγκεκριμένα, αρχικά γίνεται μετατροπή από `voxels` σε `floats`, ώστε να δημιουργηθεί ο χωρο-χρονικός Gaussian πυρήνας. Μετά εφαρμόζεται ένα φίλτρο στο βίντεο που δίνεται ως `input`. Έπειτα, δημιουργούνται φίλτρα για τις μερικές παραγώγους και υπολογίζονται οι παράγωγοι του L . Ύστερα, υπολογίζεται η εξομάλυνση και οι παράγωγοι του M . Τέλος, υπολογίζεται το Κριτήριο Γωνιότητας.

Ο κώδικας εντοπίζεται στα αρχεία `part2.py`.

Παρακάτω φαίνονται ενδεικτικά κάποια στιγμιότυπα από κάθε εικόνα.

Ανίχνευση σημείων ενδιαφέροντος για Boxing:

Frame 50:



Frame 180:



Ανίχνευση σημείων ενδιαφέροντος για Running:

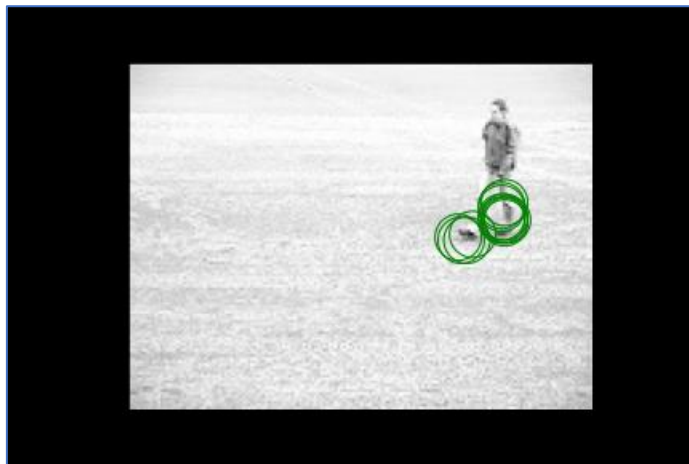
Frame 20:



Frame 100:



Ανίχνευση σημείων ενδιαφέροντος για Walking:



Frame 50:



Frame 100:

Σχολιασμός:

Σε περίπτωση που ζητείται η πολύ-κλιμακωτή προέκταση, θα προκύψει ένα σύνολο από τούπλες της μορφής $(x, y, t, s^k\sigma, s^k\tau, \widehat{H}_k(x, y, t))$ και θα διατηρηθούν μόνο οι κλίμακες που δίνουν μέγιστο εντός καθορισμένων περιοχών.

Η μέθοδος Harris βρίσκει κατά κύριο λόγο, γωνιώδη σημεία που εμφανίζονται ανάμεσα στα frames. Για παράδειγμα, είναι ορατό στα frames για το Boxing ότι έχει εντοπίσει τους αγκώνες του παλαιστή. Αντίθετα στα frames για το Running δεν έχει εντοπίσει κάποιο συγκεκριμένο σημείο αλλά γενικότερα εστιάζει στα πόδια του αθλητή. Στα frames για το Walking φαίνεται να συμβαίνει κάτι αντίστοιχο με το Running αλλά η εστίαση είναι βελτιωμένη. Τέλος, για τα σημεία που δεν υπάρχει κίνηση, δεν γίνεται ανίχνευση και επίσης εντοπίζονται μικρά σφάλματα σε κάποιες περιοχές.

2.1.2 Ζητείται ο υπολογισμός του ανιχνευτή Gabor ο οποίος βασίζεται στο χρονικό φιλτράρισμα του βίντεο με ένα ζεύγος Gabor φίλτρων αφού πρώτα αυτό έχει υποστεί εξομάλυνση στις χωρικές διαστάσεις μέσω ενός 2D γκαουσιανού πυρήνα $g(x, y; \sigma)$ με τυπική απόκλιση σ . Τα Gabor φίλτρα ορίζονται ως:

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

Για τον υπολογισμό της κρουστικής απόκρισης των Gabor θεωρήθηκε μέγεθος παραθύρου $[-2\tau, 2\tau]$ και κανονικοποιήθηκε με την L1 νόρμα.

Η συχνότητα ω του Gabor φίλτρου συνδέεται με την χρονική κλίμακα τ (απόκλιση της γκαουσιανής συνιστώσας του) μέσω της σχέσης: $\omega = 4/\tau$. Το κριτήριο σημαντικότητας προκύπτει παίρνοντας την τετραγωνική ενέργεια της εξόδου για το ζεύγος Gabor φίλτρων:

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

Ο ανιχνευτής Gabor βασίζεται στο χρονικό φιλτράρισμα του βίντεο με Gabor κυματίδια h_{ev} , h_{od} και στην εξαγωγή σημείων ενδιαφέροντος με χρήση του ακόλουθου Κριτηρίου Σημαντικότητας:

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

Το ζεύγος των Gabor φίλτρων δίνεται από τις σχέσεις:

$$h_{ev} = -\cos(2\pi t\omega) \exp(-t^2/2\tau^2) \quad \& \quad h_{od} = -\sin(2\pi t\omega) \exp(-t^2/2\tau^2)$$

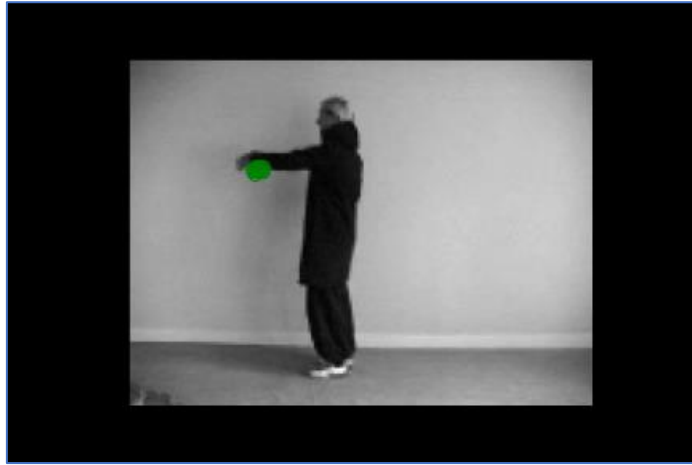
Για την εξαγωγή των ακροτάτων ακολουθείται η ίδια διαδικασία με τον ανιχνευτή Harris Stephens.

Η συνάρτηση GaborDetector υλοποιεί την εύρεση σημείων ενδιαφέροντος και επιστρέφει το Κριτήριο Σημαντικότητας για το βίντεο που δίνεται ως input. Πιο συγκεκριμένα, αρχικά γίνεται μετατροπή από voxels σε floats, ώστε να δημιουργηθεί η χωρική εξομάλυνση με Gaussian πυρήνα. Μετά δημιουργείται ένα Gabor φίλτρο. Έπειτα, γίνονται οι κατάλληλες μετατροπές για μπορέσει να γίνει η συνέλιξη. Τέλος, υπολογίζεται το Κριτήριο Σημαντικότητας.

Ο κώδικας εντοπίζεται στα αρχεία `part2.py`.

Παρακάτω φαίνονται ενδεικτικά κάποια στιγμιότυπα από κάθε εικόνα.

Ανίχνευση σημείων ενδιαφέροντος για Boxing:



Frame 50:



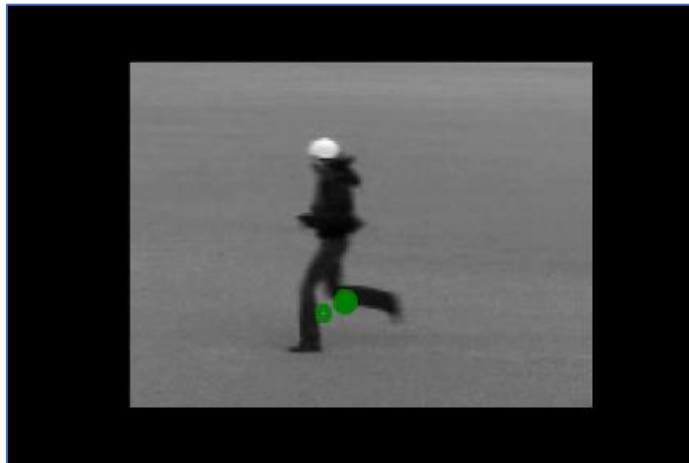
Frame 130:

Ανίχνευση σημείων ενδιαφέροντος για Running:

Frame 100:



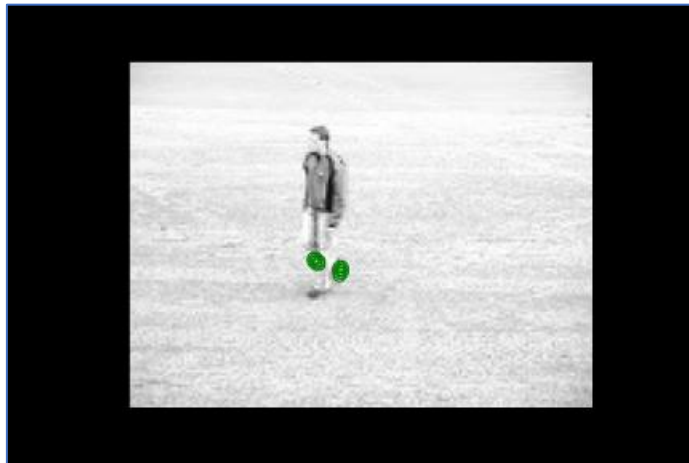
Frame 20:



Ανίχνευση σημείων ενδιαφέροντος για Walking:



Frame 50:



Frame 100:

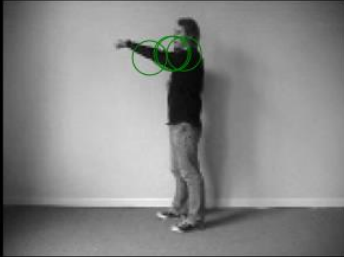



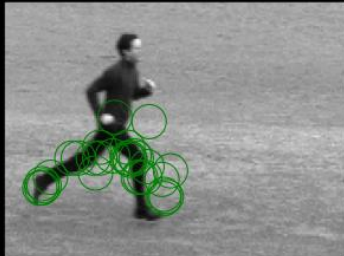

Σχολιασμός:

Η ανίχνευση με την μέθοδο Gabor φαίνεται να είναι καλύτερη από την Harris. Στα frames για το Boxing, τα σημεία ενδιαφέροντος εντοπίζονται γύρω από τα χέρια του παλαιστή. Στα frames για το Running, τα σημεία ενδιαφέροντος βρίσκονται κυρίως γύρω από την περιοχή των ποδιών. Στα frames για το Walking, τα σημεία ενδιαφέροντος μοιάζουν με εκείνα του Running. Τέλος, και εδώ για τα σημεία που δεν υπάρχει κίνηση, δεν γίνεται ανίχνευση.

2.1.3 Για καθένα ανιχνευτή υπολογίζονται τα σημεία ενδιαφέροντος σαν τα τοπικά μέγιστα του κριτηρίου σημαντικότητας. Για απλότητα, επιστρέφονται τα N σημεία με τις μεγαλύτερες τιμές του κριτηρίου σημαντικότητας (π.χ. τα 500-600 πρώτα). Ζητείται ο απεικονισμός για επιλεγμένα frames των κριτηρίων σημαντικότητας καθώς και τα σημεία που προκύπτουν χρησιμοποιώντας την συνάρτηση `show_detection` από το συμπληρωματικό υλικό. Αφήνεται στην κρίση μας ο πειραματισμός με διαφορετικές χωρικές και χρονικές κλίμακες ή και με πολλαπλές κλίμακες. Τέλος, ζητείται ο σχολιασμός ως προς τον τύπο των σημείων που ανιχνεύουν οι δύο μέθοδοι.

Βοήθεια για Python: συναρτήσεις `(cv2) getGaussianKernel`, `(scipy.ndimage) convolve1d`, `(numpy) argsort`, `unravel_index`.

Για πιο εποπτική & συνοπτική ανάλυση, φαίνονται παρακάτω οι εικόνες για Harris και Gabor.

	Harris Stephens Ανιχνευτής	Gabor Ανιχνευτής
Boxing		
Walking		
Running		

Για Harris:

Κριτήριο Σημαντικότητας για Boxing:

Frame 50:

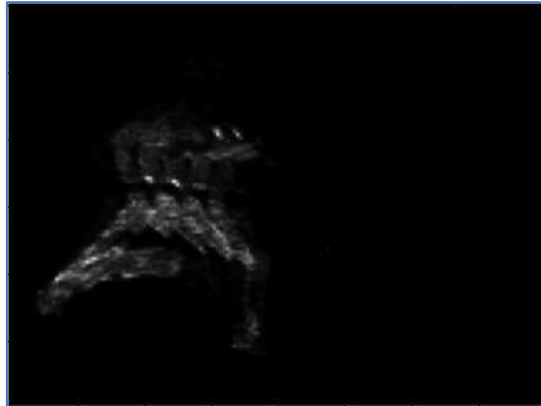


Frame 180:



Κριτήριο Σημαντικότητας για Running:

Frame 100:



Frame 20:



Κριτήριο Σημαντικότητας για Walking:

Frame 50:



Frame 100:



Για Gabor:

Κριτήριο Σημαντικότητας για Boxing:

Frame 20:



Frame 120:



Κριτήριο Σημαντικότητας για Running:

Frame 100:



Frame 20:



Κριτήριο Σημαντικότητας για Walking:

Frame 50:



Frame 100:



Στο Παράρτημα της Αναφοράς φαίνεται ενδεικτικός κώδικας με κάποια μικρά tests για πειραματισμό.

Σχολιασμός:

Για μεγάλες τιμές της χωρικής κλίμακας, παρατηρούνται ικανοποιητικά αποτελέσματα όμως καθίσταται αδύνατος ο εντοπισμός μικρών κινήσεων. Μια καλή πρακτική θα ήταν, για τα βίντεο με μεγάλη κίνηση να υπάρχουν μεγάλες κλίμακες ενώ για τα βίντεο με μικρές κινήσεις, μικρότερη κλίμακα.

Για μεγάλες τιμές της χρονικής κλίμακας, προκύπτουν σφάλματα στον Harris Stephens ανιχνευτή καθώς λαμβάνονται πληροφορίες και από τα επόμενα frames, ενώ για πολύ μικρές κινήσεις καθίσταται αδύνατος ο εντοπισμός. Όσον αφορά τον Gabor ανιχνευτή, μεταβολές της χρονικής κλίμακας μπορούν να προκαλέσουν απότομες αλλαγές στο Κριτήριο Σημαντικότητας. Επομένως, όσο μεγαλύτερες είναι οι τιμές, τόσο περισσότερα μελλοντικά frames επηρεάζουν το αποτέλεσμα ενώ, όσο πιο αργές είναι οι κινήσεις τόσο καλύτερα.

2.2 Χωρο-χρονικοί Ιστογραφικοί Περιγραφητές

Οι χωρο-χρονικοί περιγραφητές που θα χρησιμοποιηθούν βασίζονται στον υπολογισμό ιστογραμμάτων της κατευθυντικής παραγώγου (HOG) και της οπτικής ροής (HOF – Histogram of Oriented Flow) γύρω από τα σημεία ενδιαφέροντος που υπολογίστηκαν.

2.2.1 *Για κάθε frame του βίντεο ζητείται ο υπολογισμός του διανύσματος κλίσης (gradient) και η TVL_1 οπτική ροή σε κάθε pixel.*

Ο σχετικός κώδικας φαίνεται στο Part 2.2.1.

2.2.2 *Στη συνέχεια, με την χρήση της συνάρτησης `orientation_histogram` από το συμπληρωματικό υλικό του μέρους αυτού γίνεται ο υπολογισμός των 2 ιστογραφικών περιγραφητών. Η συνάρτηση αυτή δέχεται ως είσοδο το διανυσματικό πεδίο (κατευθυντικές παραγώγους είτε κατεύθυνση ροής), το μέγεθος του grid και το πλήθος των bins και επιστρέφει την ιστογραμματική περιγραφή της αντίστοιχης περιοχής. Ζητείται η εξαγωγή των διανυσματικών πεδίων που απαιτούνται για μια (τετραγωνική) περιοχή $4 \times \sigma$ γύρω από το εκάστοτε σημείο ενδιαφέροντος (από την εικόνα που αντιστοιχεί στο frame που ανιχνεύθηκαν σημεία ενδιαφέροντος). Προσοχή στα όρια της εικόνας. Η δημιουργία του HOG/HOF περιγραφητή γίνεται με την συνένωση των δύο επιμέρους περιγραφητών.*

Ο σχετικός κώδικας φαίνεται στο Part 2.2.2.

HOG

Η βασική ιδέα πίσω από την δημιουργία των HOG περιγραφητών είναι ότι η όψη του τοπικού αντικειμένου (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 ένα 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 ένα contrast-normalization.

Τα HOG Features αποτελούν ένα σύνολο χαρακτηριστικών για την περιγραφή της δομής του σχήματος σε μια εικόνα και χρησιμοποιούνται με μεγάλη επιτυχία στην αναγνώριση αντικειμένων που περιέχονται σε μια εικόνα. Τα HOGs παρέχουν μια πυκνή επικαλυπτόμενη περιγραφή των περιοχών μιας εικόνας και υπολογίζονται σε ένα πυκνό πλέγμα ομοιόμορφων cells και χρησιμεύουν στην ομαδοποίηση και κωδικοποίηση των κατευθύνσεων μιας εικόνας σε ιστογράμματα.

Για τον υπολογισμό των HOG Features πρώτα υπολογίζεται η παράγωγος της εικόνας I με χρήση Sobel Kernels για την x και y κατεύθυνση αντίστοιχα. Οι τελεστές Sobel δίνονται ως:

$$D_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, D_y = D_x^T, g_x = I * D_x, g_y = I * D_y$$

Και το μέτρο και η φάση της παραγώγου δίνονται ως εξής:

$$\nabla I(x, y) \approx (g_x(x, y), g_y(x, y)) \quad M(x, y) = \sqrt{g_x^2 + g_y^2} \quad A(x, y) = \arg(g_x, g_y)$$

Γύρω από κάθε περιοχή (x_0, y_0) ενδιαφέροντος, εκτείνεται ένα τετραγωνικό χωρίο $B(x_0, y_0)$ το οποίο μοιράζεται σε cells και χωρίζονται σε n_b bins όλες οι γωνίες στο διάστημα $0-360$ με βήμα $s = 360/n_b$ για κάθε cell. Η τιμή κάθε bin j ορίζεται ως:

$$b(I, x_0, y_0) = \sum_{(u,v) \in C(x_0, y_0)} 1 \left\{ \left\lfloor \frac{A(u,v)}{s} \right\rfloor = j \right\} M(u, v)$$

Έπειτα, εφαρμόζεται κανονικοποίηση με την L2 Νόρμα: $\hat{b}(j, x_0, y_0) = \frac{b(j, x_0, y_0)}{(\sum_{k=0}^{n_b-1} b^2(k, x_0, y_0))^{1/2}}$

Τέλος, για κάθε παράθυρο, εφαρμόζεται concat στα HOG όλων των cells και λαμβάνονται τα HOG Features για το συγκεκριμένο σημείο. Με αυτό τον τρόπο τα HOG αποκαλύπτουν την πληροφορία για το τοπικό σχήμα ενώ παράλληλα κωδικοποιούν τις κατευθύνσεις της κλίσης της εικόνας στο ιστόγραμμα.

Σχολιασμός:

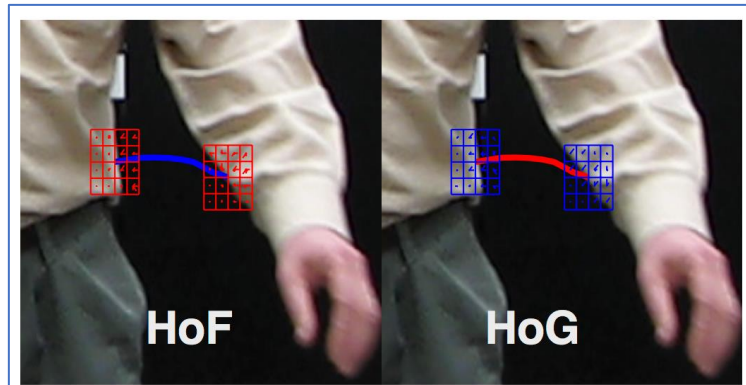
Τα HOG είναι αρκετά αμετάβλητα στις αλλαγές της φωτεινότητας καθώς εφαρμόζουν τοπική κανονικοποίηση των ιστογραμμάτων σε επικαλυπτόμενες περιοχές.

HOF

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 έχουν μεγαλύτερη ακρίβεια για την εξαγωγή για των χαρακτηριστικών σε σχέση με τους HOG και για τον λόγο αυτό οι HOG χρησιμοποιούνται περισσότερο.

Τα HOF Features εξάγονται με την ίδια διαδικασία με τους HOG περιγραφητές με την μόνη διαφορά ότι στην θέση των g_x, g_y υπάρχει το διανυσματικό πεδίο της οπτικής ροής d_x, d_y .

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



HOG/HOF

Ο ιστογραφικός περιγραφητής HOG/HOF αποτελεί ένα concatenation του HOG & HOF περιγραφητή γύρω από το σημείο ενδιαφέροντος (x, y, t) . Κάθε βίντεο επομένως έχει ένα διάνυσμα χαρακτηριστικών $k \times (2n_b)$ όπου $k \in [500, 600]$.

HOG3D

Με τις παραπάνω τεχνικές, είναι δυνατό να επεκταθεί το HOG σε 3 διαστάσεις και να κατασκευαστεί ο ιστογραφικός περιγραφητής χρησιμοποιώντας το μέτρο $M(x, y) = \sqrt{g_x^2 + g_y^2 + g_t^2}$ και τις δύο γωνίες $\theta(x, y, t) = \arg(g_x, g_y)$ & $\phi(x, y, t) = \arg(g_y, g_t)$ ώστε να κατασκευαστεί ένα ιστογραφικός περιγραφητής διαστάσεων $k \times n_b^2$ με παρόμοια λογική με τον απλό HOG.

Σχολιασμός:

Στα πειράματα δεν χρησιμοποιείται ο περιγραφητής HOG3D καθώς έχει ήδη βρει περιοχή ενδιαφέροντος λαμβάνοντας υπόψιν την χρονική πληροφορία είτε με χρήση της χωρο-χρονικής εκδοχής του Harris Stephens είτε μέσω του ανιχνευτή Gabor.

Οι υλοποιήσεις φαίνονται στο αρχείο `part2.py`.

Η συνάρτηση `HOG_HOF_Descriptor` είναι εκείνη που υπολογίζει ανάλογα με την παράμετρο `descriptor`, αν θα υπολογίσει HOG, HOF, HOG/HOF.

2.3 Κατασκευή Bag of Visual Words και χρήση Support Vector Machines για την ταξινόμηση δράσεων

Στο ερώτημα αυτό γίνεται κατηγοριοποίηση των βίντεο με τις ανθρώπινες δράσεις σε 3 κατηγορίες/κλάσεις (που η κάθε μια αντιπροσωπεύει ένα διαφορετικό είδος δράσης) με χρήση των BoVW αναπαραστάσεων βασισμένων στα HOG/HOF χαρακτηριστικών, που εξάγονται στα προηγούμενα ερωτήματα. Το τελικό αποτέλεσμα είναι το ποσοστό επιτυχούς ταξινόμησης δράσεων, χρησιμοποιώντας SVM ταξινομητή.

2.3.1 Ζητείται ο διαχωρισμός του συνόλου των βίντεο σε σύνολο εκπαίδευσης (train set) και σύνολο δοκιμής (test set) με βάση το αρχείο που δίνεται στο συμπληρωματικό υλικό.

Η συνάρτηση `division_of_data` είναι εκείνη που διαχωρίζει τα δεδομένα σε train set & test set.

2.3.2 Ζητείται ο υπολογισμός της τελικής αναπαράστασης (global representation) για κάθε βίντεο με την bag of visual words (BoVW) τεχνική που περιγράφεται στην 1^η εργαστηριακή άσκηση, χρησιμοποιώντας μόνο τα βίντεο εκπαίδευσης. Για τον υπολογισμό των BoVW ιστογραμμάτων γίνεται χρήση της συνάρτησης `bag_of_words` από το συμπληρωματικό υλικό αυτού του μέρους.

Η συνάρτηση `BoVW_implem` είναι εκείνη που με την χρήση της `descriptor_for_video_set` υπολογίζει για κάθε βίντεο την bag of visual words.

2.3.3 Το τελικό στάδιο συνίσταται στην τελική κατηγοριοποίηση των εικόνων με βάση την BoVW αναπαράσταση. Για την κατηγοριοποίηση χρησιμοποιείται ένας SVM ταξινομητής κατάλληλα προσαρμοσμένος για πολλαπλές κλάσεις. Η όλη διαδικασία υλοποιείται με τη συνάρτηση `svm_train_test` από το συμπληρωματικό υλικό, η οποία δέχεται 2 numpy πίνακες διαστάσεων $N_{train} \times D$ και $N_{test} \times D$, όπου N_{train} , N_{test} ο αριθμός των βίντεο εκπαίδευσης και δοκιμής αντίστοιχα και D η διάσταση του κάθε BoVW διανύσματος. Η συνάρτηση επιστρέφει το αποτέλεσμα της αναγνώρισης καθώς και το συνολικό ποσοστό επιτυχίας.

Η συνάρτηση `SVM_classification` είναι εκείνη που κατηγοριοποιεί τα δεδομένα με βάση έναν SVM ταξινομητή.

2.3.4 Συνίσταται ο πειραματισμός με τους διαφορετικούς συνδυασμούς ανιχνευτών/περιγραφητών ώστε να παρατηρηθούν οι μεταβολές στην κατηγοριοποίηση και να αναφερθεί ο καλύτερος συνδυασμός.

Ένας μέσος όρος αποτελεσμάτων που προκύπτουν για κάποιους διαφορετικούς συνδυασμούς, φαίνεται παρακάτω:

HOG με Harris Stephens	~ 90 %
HOF με Harris Stephens	~ 80 %
HOG/HOF με Harris Stephens	~ 90 %
HOG με Gabor	~ 90 %
HOF με Gabor	~ 80 %
HOG/HOF με Gabor	~ 90 %

Σχολιασμός:

Στα αποτελέσματα εμφανίζεται τυχαιότητα αφού τα ποσοστά μπορεί να διαφοροποιούνται κάθε φορά που τρέχει κανείς τον κώδικα.

Ο HOG δεν δίνει πάντα τα επιθυμητά αποτελέσματα αφού δεν μπορεί να ξεχωρίσει τις κινήσεις σε όλες τις περιπτώσεις όμως, παραμένει αμετάβλητος σε αλλαγές φωτεινότητας και σε περιστροφικές κινήσεις ή μετατοπίσεις κλίμακας.

Ο HOF δίνει σχετικά καλύτερα αποτελέσματα καθώς εστιάζει στην κίνηση επειδή λαμβάνει για γειτονικές περιοχές την οπτική ροή για να την απεικονίσει στα ιστογράμματα και έτσι επιτυγχάνει τον διαχωρισμό των βίντεο που αφορούν boxing από τα άλλα

Τέλος, οι HOG/HOF δίνουν τα καλύτερα αποτελέσματα και για τους δύο ανιχνευτές σημείων αφού συνδυάζουν τις δύο προηγούμενες μεθόδους και εξετάζει ταυτόχρονα και την παράγωγο αλλά και την οπτική ροή.

2.3.5 Συνίσταται ο πειραματισμός με διαφορετικούς διαμερισμούς των δεδομένων σε *train* και *test set* και ο σχολιασμός της επίδρασης που έχουν στα αποτελέσματα.

Σχολιασμός:

Όσον αφορά την απόδοση, αυτή αξιολογείται μέσα από διάφορα πειράματα με πιθανούς συνδυασμούς για *test & train*.

Φάνηκε πως ο βέλτιστος συνδυασμός descriptor-detector είναι ο Gabor-HOG & Gabor-HOG/HOF. Επιβεβαιώνεται δηλαδή και πειραματικά ότι ο detector με τα καλύτερα αποτελέσματα είναι ο Gabor.

Αξίζει να σημειωθεί ότι ο HOG επειδή δεν επηρεάζεται από περιστροφές και κλιμακώσεις φαίνεται να είναι καλύτερος από τον HOF. Ωστόσο, ο detector HOG/HOF φαίνεται να οδηγεί στα καλύτερα δυνατά αποτελέσματα είτε με Gabor είτε με Harris Stephens.

Υπογραμμίζεται ότι παρατηρούνται και αρκετές εσφαλμένες κατηγοριοποιήσεις μεταξύ *running* και *walking* το οποίο είναι λογικό και από την πραγματικότητα αφού οι δυο αυτές ενέργειες είναι αρκετά παρόμοιες σε αντίθεση με το *boxing*.

Για διαφορετικά δεδομένα στους πειραματισμούς οδηγείται κανείς στα εξής συμπεράσματα. Για μεγάλα *train set* αναμένεται η ύπαρξη μεγάλων ποσοστών. Αντιθέτως, για μικρό *train set* είναι αναμενόμενο να προκύψουν χαμηλά ποσοστά. Τέλος, για μειωμένα *train set* κάποιας συγκεκριμένης δραστηριότητας, τα ποσοστά ακρίβειας αναμένεται να είναι μειωμένα.

Μια ενδεικτική υλοποίηση φαίνεται στο Παράρτημα της Αναφοράς.

Παράρτημα:

Μέρος 1.2.2:

```
#####  
# Motion Tracking in all Frames  
  
def motion_tracking(head_dict, lefthand_dict, righthand_dict, rho, epsilon, savedir,  
multiscale = False, scale=0):  
    I1 = np.array(cv2.imread('GreekSignLanguage/1.png', 1))  
  
    faceOld = I1[head_dict['y']:head_dict['y']+head_dict['height'], head_dict['x']:h  
ead_dict['x']+head_dict['width']]  
    leftOld = I1[lefthand_dict['y']:lefthand_dict['y']+lefthand_dict['height'], left  
hand_dict['x']:lefthand_dict['x']+lefthand_dict['width']]  
    rightOld = I1[righthand_dict['y']:righthand_dict['y']+righthand_dict['height'],  
righthand_dict['x']:righthand_dict['x']+righthand_dict['width']]  
    plot_boxes(I1, head_dict, lefthand_dict, righthand_dict, True ,directory=savedir  
, name='1')  
  
    noofimages=67  
    for i in range(2,noofimages):  
        path = 'GreekSignLanguage/'+str(i)+'.png'  
        newimage = np.array(cv2.imread(path, 1))  
  
        faceNew = newimage[head_dict['y']:head_dict['y']+head_dict['height'], head_d  
ict['x']:head_dict['x']+head_dict['width']]  
        leftNew = newimage[lefthand_dict['y']:lefthand_dict['y']+lefthand_dict['heig  
ht'], lefthand_dict['x']:lefthand_dict['x']+lefthand_dict['width']]  
        rightNew = newimage[righthand_dict['y']:righthand_dict['y']+righthand_dict['  
height'], righthand_dict['x']:righthand_dict['x']+righthand_dict['width']]  
  
        corners_head = shi_tomasi(faceOld, parameters, False)  
        corners_left = shi_tomasi(leftOld, parameters, False)  
        corners_right = shi_tomasi(rightOld, parameters, False)  
  
        if multiscale==True:  
            dx1, dy1 = multi_lk(faceOld, faceNew, corners_head, rho, epsilon, scale,  
parameters)  
            dx2, dy2 = multi_lk(leftOld, leftNew, corners_left, rho, epsilon, scale,  
parameters)  
            dx3, dy3 = multi_lk(rightOld, rightNew, corners_right, rho, epsilon, sca  
le, parameters)  
        else:  
            dx1, dy1 = lucas_kanade(faceOld, faceNew, corners_head, rho, epsilon, 0,  
0)  
            dx2, dy2 = lucas_kanade(leftOld, leftNew, corners_left, rho, epsilon, 0,  
0)
```

```

        dx3, dy3 = lucas_kanade(rightOld, rightNew, corners_right, rho, epsilon,
0, 0)

    displ_x1, displ_y1 = displ(dx1, dy1)
    displ_x2, displ_y2 = displ(dx2, dy2)
    displ_x3, displ_y3 = displ(dx3, dy3)

    head_dict['x'] = int(head_dict['x'] - displ_x1)
    head_dict['y'] = int(head_dict['y'] - displ_y1)

    lefthand_dict['x'] = int(lefthand_dict['x'] - displ_x2)
    lefthand_dict['y'] = int(lefthand_dict['y'] - displ_y2)

    righthand_dict['x'] = int(righthand_dict['x'] - displ_x3)
    righthand_dict['y'] = int(righthand_dict['y'] - displ_y3)

    plot_boxes(newimage, head_dict, lefthand_dict, righthand_dict, save=True, di
rectory=savedir, name=str(i))

    faceOld = np.array(faceNew)
    leftOld = np.array(leftNew)
    rightOld = np.array(rightNew)

# given Values

head_dict = {'x':138, 'y':88, 'width':73, 'height':123}
lefthand_dict = {'x':47, 'y':243, 'width':71, 'height':66}
righthand_dict = {'x':162, 'y':264, 'width':83, 'height':48}

motion_tracking(head_dict, lefthand_dict, righthand_dict, rho, epsilon, savedir = './
motion_tracking/', multiscale = False, scale=0)

```

Μέρος 1.2.3:

```

# Use given Values
head_dict = {'x':138, 'y':88, 'width':73, 'height':123}
lefthand_dict = {'x':47, 'y':243, 'width':71, 'height':66}
righthand_dict = {'x':162, 'y':264, 'width':83, 'height':48}

motion_tracking(head_dict, lefthand_dict, righthand_dict, rho, epsilon, savedir = './
multi_tracking/', multiscale = True, scale=4)

```

Μέρος 2.1.3:

```
#####
# Harris-Stephens Detector Experiment

# experiment for walking

walk_harris_stephens_criterion = HarrisStephensDetector(walking_video3, 4, 1.5, 0.005, 2)
walk_harris_stephens_points = calculate_interest_points(walk_harris_stephens_criterion, 600, 4)
show_detection(walking_video3, walk_harris_stephens_points, save_path=r"./HarrisDetection/walk")
!ffmpeg -i ./HarrisDetection/walk/frame%d.png -c:v libx264 -vf fps=25 ./HarrisDetection/harris_walk.mp4

# experiment for running

run_harris_stephens_criterion = HarrisStephensDetector(running_video3, 4, 1.5, 0.005, 2)
run_harris_stephens_points = calculate_interest_points(run_harris_stephens_criterion, 600, 4)
show_detection(running_video3, run_harris_stephens_points, save_path=r"./HarrisDetection/run")
!ffmpeg -i ./HarrisDetection/run/frame%d.png -c:v libx264 -vf fps=25 ./HarrisDetection/harris_run.mp4

# experiment for boxing

box_harris_stephens_criterion = HarrisStephensDetector(boxing_video3, 4, 1.5, 0.005, 2)
box_harris_stephens_points = calculate_interest_points(box_harris_stephens_criterion, 600, 4)
show_detection(boxing_video3, box_harris_stephens_points, save_path=r"./HarrisDetection/box")
!ffmpeg -i ./HarrisDetection/box/frame%d.png -c:v libx264 -vf fps=25 ./HarrisDetection/harris_box.mp4

# Gabor Detector Experiment

# experiment for walking

walk_gabor_criterion = GaborDetector(walking_video3, 1.6, 1.5)
walk_gabor_points = calculate_interest_points(walk_gabor_criterion, 600, 1.6)
show_detection(walking_video3, walk_gabor_points, save_path=r"./GaborDetection/walk")
!ffmpeg -i ./GaborDetection/walk/frame%d.png -c:v libx264 -vf fps=25 ./GaborDetection/gabor_walk.mp4
```

```

run_gabor_criterion = GaborDetector(running_video2, 1.6, 1.5)
run_gabor_points = calculate_interest_points(run_gabor_criterion, 600, 1.6)
show_detection(running_video2, run_gabor_points, save_path=r"./GaborDetection/run")
!ffmpeg -i ./GaborDetection/run/frame%d.png -c:v libx264 -
vf fps=25 ./GaborDetection/gabor_run.mp4

# experiment for boxing
box_gabor_criterion = GaborDetector(boxing_video1, 1.6, 1.5)
box_gabor_points = calculate_interest_points(box_gabor_criterion, 600, 1.6)
show_detection(boxing_video1, box_gabor_points, save_path=r"./GaborDetection/box")
!ffmpeg -i ./GaborDetection/box/frame%d.png -c:v libx264 -
vf fps=25 ./GaborDetection/gabor_box.mp4

```

Μέρος 2.3:

Αναλυτική Περιγραφή του 2.3:

Αρχικά, εξάγονται οι περιγραφητές HOG/HOF από το σύνολο των σημείων ενδιαφέροντος που έχουν ανιχνευτεί με τους ανιχνευτές σημείων ενδιαφέροντος για κάθε βίντεο.

Έπειτα, χρησιμοποιείται η μέθοδος Bag of Visual Words για να προκύψει ένα embedding για όλο το βίντεο από το σύνολο των features vectors που έχουν εξαχθεί. Ειδικότερα, επιλέγονται τα features vectors όλων των βίντεο και εκπαιδεύεται ένα k-means clustering. Έστερα, κάθε βίντεο αντιστοιχίζεται σε ένα από τα clusters και έτσι προκύπτει μια διανυσματική αναπαράσταση για όλο το βίντεο, γνωστή ως video embedding. Η διαδικασία εξαγωγής των clusters αντιστοιχεί στην δημιουργία του πίνακα $\mathbb{F} = (F_1 F_2 \dots F_N)^T$ που έχει όλα τα χαρακτηριστικά vertically stacked. Μετά, λαμβάνεται ένα δείγμα $[N/2]$ γραμμών του πίνακα, έστω $\hat{\mathbb{F}}$, οι γραμμές του οποίου χρησιμοποιούνται ως inputs στον αλγόριθμο K-means για την εξαγωγή των κέντρων n_c clusters, έστω z_1, \dots, z_{n_c} . Έπειτα, για κάθε διάνυσμα χαρακτηριστικών, υπολογίζεται η κατανομή

$$N(k, j) = |\{f \in \text{row}(\mathbb{F}_k) \mid j = \arg \min_{r=1, \dots, n_c} \{\|f - z_r\|_2\}\}|$$

Εν συνεχεία, γίνεται κανονικοποίηση με την L2 νόρμα $x(k, j) = \frac{N(k, j)}{(\sum_{r=1}^{n_c} N^2(k, r))^{1/2}}$

Το διάνυσμα $x_k = (x(k, j))_{1 \leq j \leq n_c}$ αποτελεί την διανυσματική αναπαράσταση (embedding) του βίντεο που αριθμείται με το δείκτη k.

Εφόσον πλέον έχει ολοκληρωθεί η εξαγωγή των διανυσματικών αναπαραστάσεων για κάθε βίντεο, σκοπός είναι να γίνει η unsupervised ανίχνευση δραστηριοτήτων. Αυτό επιτυγχάνεται κατασκευάζοντας το δένδrogram που προκύπτει από τη διαδικασία συνένωσης των διανυσματικών αναπαραστάσεων για το βίντεο σε βήματα έως ότου καταλήξει σε ένα μοναδικό cluster για όλα τα βίντεο.

Τα βίντεο χωρίζονται σε τρεις διαφορετικές ανθρώπινες δραστηριότητες (boxing, walking, running) οπότε είναι αναμενόμενο το δένδροδιάγραμμα να χωριστεί σε τρεις διακριτές ομάδες και τα στοιχεία να έχουν μικρή απόσταση μεταξύ τους σε σχέση με τα στοιχεία άλλων ομάδων.

Ο πειραματισμός περιλαμβάνει διαφορετικά affinities, δηλαδή διαφορετικά metrics για τις αποστάσεις των διανυσμάτων, πέρα από την χ^2 απόσταση, όπως την ευκλείδεια απόσταση, την minkowski κλπ. Επίσης, γίνεται πειραματισμός ως προς τον τρόπο που ενώνονται τα clusters στα διαδοχικά βήματα, όπως οι μέθοδοι single, average, median κλπ. Τέλος, ο πειραματισμός γίνεται με διαφορετικό αριθμό από clusters για την εξαγωγή των BoVW για τα βίντεο. Από αυτά τα πειράματα, συμπεραίνει κανείς ότι τα καλύτερα αποτελέσματα τα δίνει η περίπτωση όπου γίνεται καλός διαχωρισμός των διαφορετικών δραστηριοτήτων με χρήση απόστασης χ^2 , Gabor φίλτρων, μεθόδου average και με αριθμό clusters 50.

Σχολιασμός:

Η αλλαγή της μεθόδου σχηματισμού των clusters αλλάζει σημαντικά τα αποτελέσματα. Ο διαχωρισμός με το boxing είναι ιδιαίτερα καθαρός αλλά τα walking, running έχουν σημαντική επικάλυψη. Επιπλέον, παρατηρείται πως ο μικρότερος αριθμός clusters χειροτερεύει το αποτέλεσμα. Ακόμη, μέτρια αποτελέσματα φαίνονται και με τη χρήση του Harris Stephens ανιχνευτή.

Μέρος 2.3.5:

```
#####
# Part 2.3.5 - Experiments with different data partitions
#####
# Implementation of experiments
# (does not run because division_of_data needs modified txt input)
# Alterations can be tested using little train set, big train set etc.

train_set, test_set, train_tags, test_tags = division_of_data('____.txt')
sigma = typ['sigma']
tau = typ['tau']
k = typ['k']
s = typ['s']
num_centers = 20

# Detector: Harris, Descriptor: HOG
print('Classification 1')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOG', 'Harris', sigma, tau, k, s, 20)
print()

# Detector: Harris, Descriptor: HOF
print('Classification 3')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOF', 'Harris', sigma, tau, k, s, 20)
print()
```

```

# Detector: Harris, Descriptor: HOG/HOF
print('Classification 5')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOG/HOF', 'Harris',
sigma, tau, k, s, 20)
print()

# Detector: Gabor, Descriptor: HOG
print('Classification 2')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOG', 'Gabor', sigma
, tau, k, s, 20)
print()

# Detector: Gabor, Descriptor: HOF
print('Classification 4')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOF', 'Gabor', sigma
, tau, k, s, 20)
print()

# Detector: Gabor, Descriptor: HOG/HOF
print('Classification 6')
SVM_classification(train_set, test_set, train_tags, test_tags, 'HOG/HOF', 'Gabor', s
igma, tau, k, s, 20)
print()

```

Γενικότερα συμπεράσματα:

Από τους διάφορους πειραματισμούς παρατηρείται ότι ο διαχωρισμός του boxing από τα άλλα δύο είναι αρκετά εύκολος καθώς διαφέρει από τις άλλες ενέργειες. Οι δραστηριότητες walking και running παρουσιάζουν αρκετές ομοιότητες και συχνά και αυτό φαίνεται και από τις κοντινές τιμές των clusters στις δύο περιπτώσεις. Επισημαίνεται, ακόμη, ότι ο αριθμός 50 για τα clusters λειτουργεί σε αρκετές περιπτώσεις αρκετά καλά καθώς είναι αρκετά πολλά ώστε οι αποστάσεις να είναι λογικές αλλά όχι τόσο πολλά ώστε τα walking και running να επικαλυφθούν. Παρατηρείται, επίσης, ότι η απόσταση χ^2 είναι αποτελεσματικότερη στις περισσότερες περιπτώσεις. Τέλος, όσον αφορά τους ανιχνευτές, προκύπτει πως καλύτερα αποτελέσματα εμφανίζει ο ανιχνευτής Gabor ιδιαίτερα χάρη στην ικανότητά του να μην μπερδεύει τις διαφορετικές δραστηριότητες σχηματίζοντας “μεικτά” clusters.

Βιβλιογραφία - Αναφορές

- [1] Όραση Υπολογιστών – Maragos_CV_Book.pdf & παλαιότερο υλικό
- [2] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In BMVC 2008-19th British Machine Vision Conference, pages 275–1. British Machine Vision Association, 2008.
- [3] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.