

Όραση Υπολογιστών

2η Εργαστηριακή Άσκηση

Ομαδα:

Δωροθέα Κουμίδου 03119712

Γιώργος Χαραλάμπους 03119706



Σχολή Ηλεκτρολόγων μηχανικών &
Μηχανικών Υπολογιστών

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

Στόχος του πρώτου μέρους είναι η υλοποίηση ενός συστήματος παρακολούθησης προσώπου και χεριών, που περιλαμβάνει α) την ανίχνευση των περιοχών ενδιαφέροντος και β) την παρακολούθησή τους σε μία ακολουθία από frames μέσω των υπολογιζόμενων διανυσμάτων οπτικής ροής (από τον αλγόριθμο Lukas Kanade)

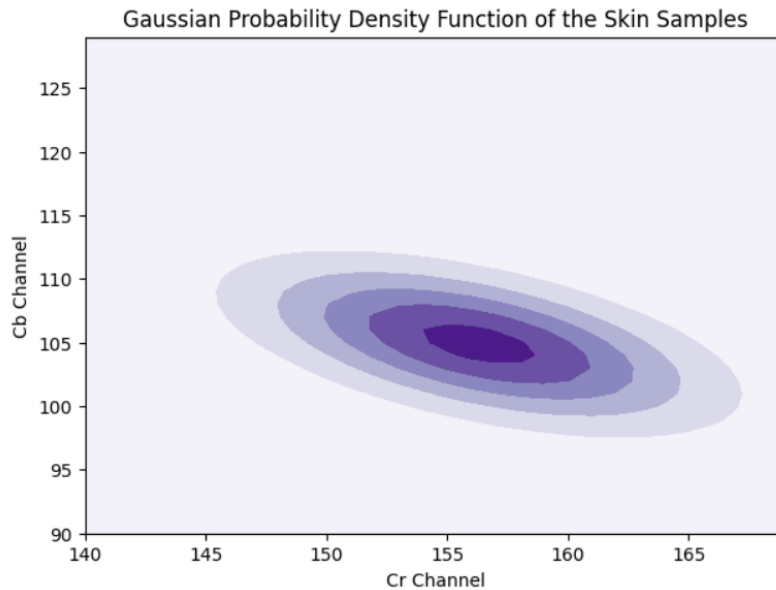
1.1 Ανίχνευση Δέρματος και Προσώπου

Σε αυτό το ερώτημα ζητείται η ανίχνευση των σημείων δέρματος της παρακάτω εικόνας:



Χρησιμοποιώντας τον χρωματικό χώρο YcbCr (συγκεκριμένα μόνο τις συνιστώσες Cb, Cr που περιγράφουν την ταυτότητα του χρώματος) και το δοθέν αρχείο skinSamplesRGB.mat λαμβάνονται τα διανύσματα μέσης τιμής και συνδιακύμανσης που μοντελοποιούν το χρώμα του δέρματος. Παρακάτω φαίνονται οι τιμές των διανυσμάτων και η συνάρτηση πυκνότητας πιθανότητας της Gaussian κατανομής.

Mean = [156.3186507936508, 104.87420634920635]
 Cov = [[34.87142652 -12.72051913]
 [-12.72051913 15.65665876]]

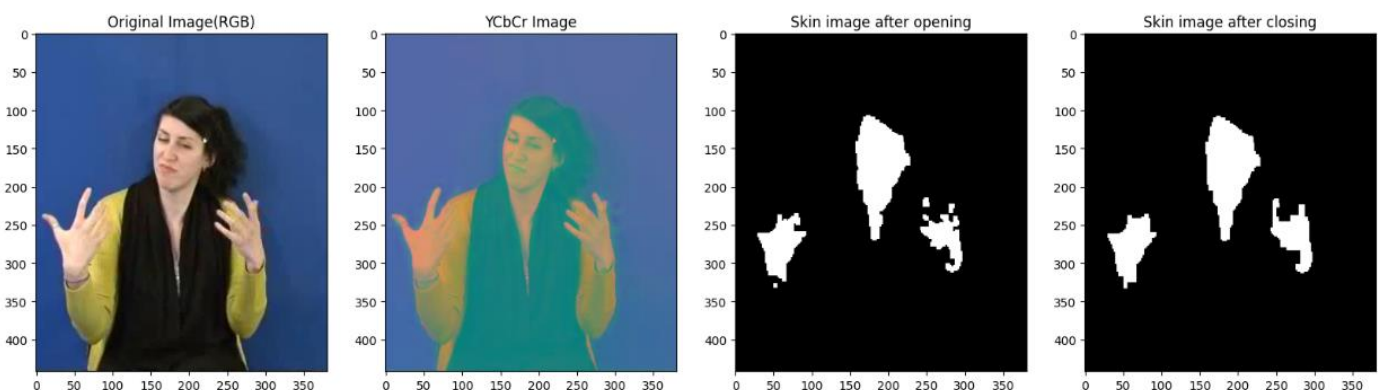


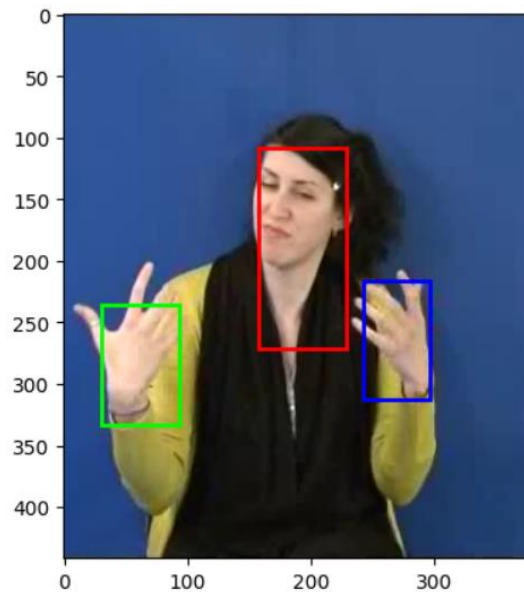
Στη συνέχεια υπολογίζεται η πιθανότητα $P(\mathbf{c}=\text{skin})$ μέσω της σχέσης:

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

και κατόπιν γίνεται κατωφλιοποίηση με κατώφλια [0.00005, 0.25] μέσω της εντολής : `cv2.threshold(p,0.00005,0.25, cv2.THRESH_BINARY)`. Έπειτα, εφαρμόζεται `opening` με [5x5] kernel και `closing` με [12x12] kernel, για την δημιουργία της τελικής συνεκτικής εικόνας η οποία περιλαμβάνει τα skin areas. Τέλος, η εντολή `add_patch` χρησιμοποιείται για την οπτική απεικόνιση των `bounding_boxes` στην αρχική εικόνα, ενώ ταυτόχρονα αποθηκεύονται οι ζητούμενες τιμές [x,y,height,width] για κάθε περιοχή δέρματος. Παρακάτω παρουσιάζονται τα ενδιάμεσα και τελικά αποτελέσματα της διαδικασίας για την πρώτη εικόνα του δοθέντος set από frames:

Detected features: 3
 Head Coords: [158, 108, 71, 163]
 Right Hand Coords: [243, 216, 54, 97]
 Left Hand Coords: [30, 236, 63, 97]





Η παραπάνω διαδικασία υλοποιείται μέσω της συνάρτησης $fd(I, \mu, cov)$, που δέχεται σαν είσοδο την εικόνα, το διάνυσμα μέσω των όρων και τον πίνακα συνδιακύμανσης, και επιστρέφει τις συντεταγμένες του bounding box για κάθε εντοπισμένο μέρος της εικόνας. Παρατηρείται ότι η συνάρτηση εντοπίζει με επιτυχία τις τρεις περιοχές δέρματος (πρόσωπο και χέρια) και δίνει ικανοποιητικά bounding boxes και τις αντίστοιχες συντεταγμένες τους.

1.2 Παρακολούθηση Προσώπου και Χεριών

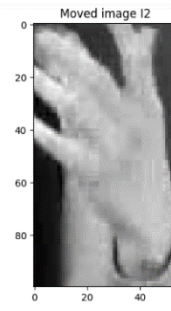
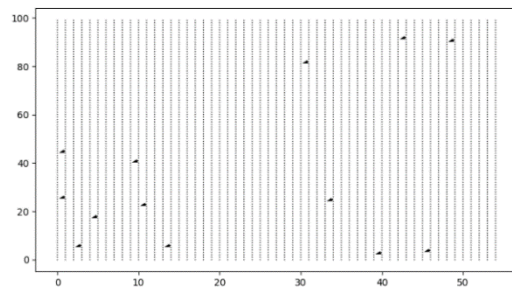
Οι τιμές των αρχικοποιήσεων που δίνονται είναι λανθασμένες, επομένως η αρχικοποίηση των bounding boxes γίνεται με βάση τις συντεταγμένες του προηγούμενου ερωτήματος.

```
face_coord = [154,102,67,115]
right_hand_coord = [243, 216, 55, 100]
left_hand_coord = [30, 220, 70, 97]
```

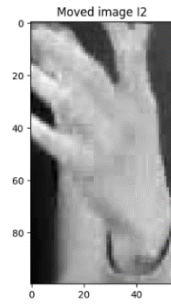
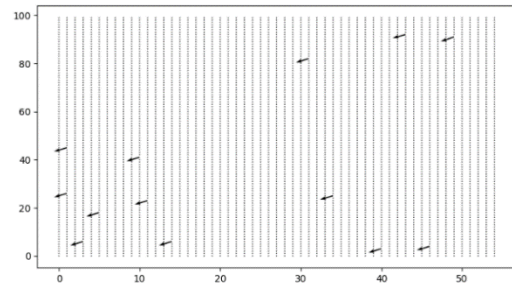
Σε αυτό το ερώτημα υλοποιείται η συνάρτηση $lk(I1, I2, features, rho, epsilon, d x0, d y0)$, που δέχεται ως είσοδο δύο εικόνες-διαδοχικά frames, τα σημεία ενδιαφέροντος, τις παραμέτρους ρ και ϵ καθώς και τις αρχικές εκτιμήσεις μας για την οπτική ροή. Η συνάρτηση εφαρμόζει τον αλγόριθμο Lucas-Kanade πάνω στα σημεία ενδιαφέροντος, και τερματίζει μετά από 100 επαναλήψεις. Αξίζει να σημειωθεί ότι η εξαγωγή των σημείων ενδιαφέροντος γίνεται με την εντολή `cv2.goodFeaturesToTrack()` των Shi και Tomashi και η οπτικοποίηση των αποτελεσμάτων με την εντολή `quiver`. Δημιουργήθηκε και δεύτερο κριτήριο τερματισμού: $criterion1 = (np.abs(dx-dx_prev) > 1e-3) \text{ or } (np.abs(dy-dy_prev) > 1e-3)$, κατά το οποίο για μικρές μετατοπίσεις είχαμε γρηγορότερη σύγκλιση.

Η υπορουτίνα δοκιμάζεται για διάφορες μετατοπίσεις μεταξύ των δύο διαδοχικών εικόνων ([1,2,5,10,15]), ώστε να εξετάσουμε τα όρια του και την ικανότητα του Lucas-Kanade να υπολογίζει καλά την οπτική ροή. Τα αποτελέσματα για το αριστερό χέρι φαίνονται παρακάτω με $\rho=5$ και $\epsilon=0.01$:

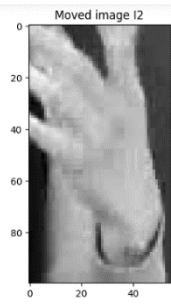
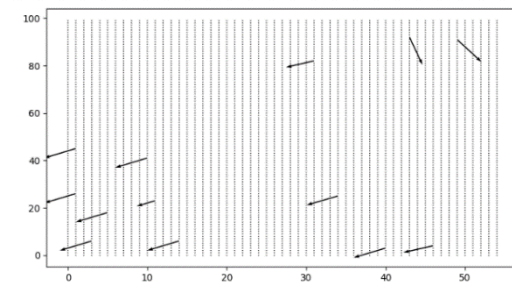
$$dx = 1, dy = 1$$



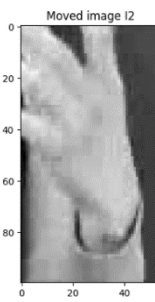
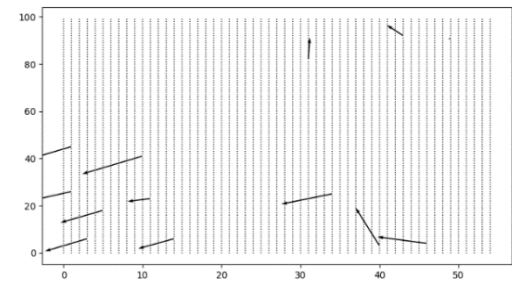
$$dx = 2, dy = 2$$



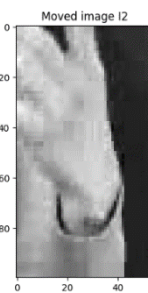
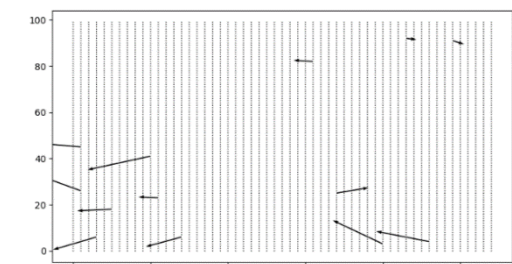
$$dx = 5, dy = 5$$



$$dx = 10, dy = 10$$



$$dx = 15, dy = 15$$



Παρατηρείται ότι για μικρές μετατοπίσεις μεταξύ των εικόνων(μέχρι $dx=5, dy=5$), η οπτική ροή των σημείων υπολογίζεται αρκετά καλά, ενώ για μεγάλες μετατοπίσεις ο αλγόριθμος δεν λειτουργεί καλά.

Για τον υπολογισμό της μετατόπισης του bounding box, δημιουργείται η συνάρτηση **displ(dx,dy)**, η οποία επιστρέφει τον μέσο όρο των οπτικών ροών των σημείων που έχουν ενέργεια μεγαλύτερη από ένα κατώφλι (εδώ επιλέγεται 0.2). Με την βοήθεια της, επιτυγχάνεται η μετακίνηση του εκάστοτε bounding box ανάλογα με την υπολογισμένη οπτική ροή. Η συνάρτηση *simple_tracker* (*limit, flow_type=True*), δέχεται ως είσοδο τον αριθμό των frames και μια Boolean μεταβλητή για την επιλογή της μεθόδου υπολογισμού οπτικής ροής(Lucas-Kanade ή TV-L1) και επιστρέφει την θέση των bounding boxes κατά τη διάρκεια της κίνησης.

Παρακάτω μερικά από τα αποτελέσματά μας, με χρήση του **Lucas-Kanade**:

Image: 1
Approximate movement of bounding box: 3.685821844184611 -13.063862888417963

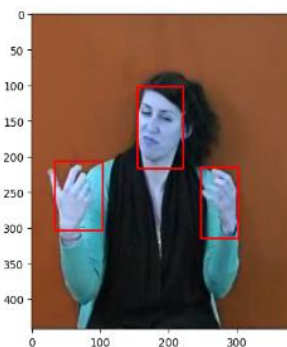


Image: 5
Approximate movement of bounding box: -2.5865699679866444 1.1524628286253329



Image: 2
Approximate movement of bounding box: 9.48767589668172 -8.07071295586549189

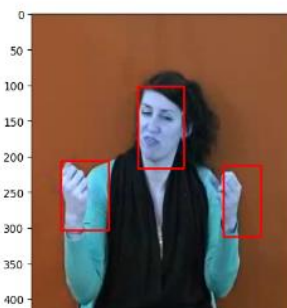


Image: 6
Approximate movement of bounding box: -5.9160847816685155 4.567058668358372

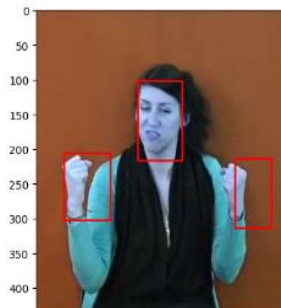


Image: 3
Approximate movement of bounding box: 4.431821718126264 -4.5388214561960805

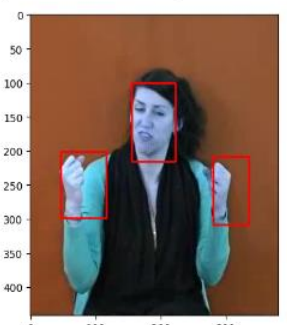


Image: 7
Approximate movement of bounding box: 1.388343612586703 -1.2182848434155022

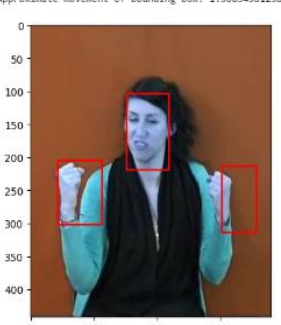


Image: 4
Approximate movement of bounding box: 3.6219668822889776 -1.7289234978668792

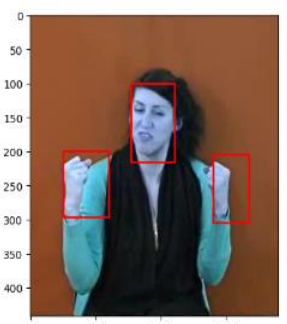
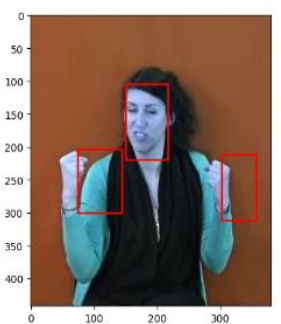


Image: 8
Approximate movement of bounding box: 31.562227588947636 -1.499342485986381



Παρακάτω μερικά από τα αποτελέσματα με χρήση της **DualTVL1OpticalFlow_create**:

Image: 1
Approximate movement of bounding box: 1.139611383192133 -2.351522141177183

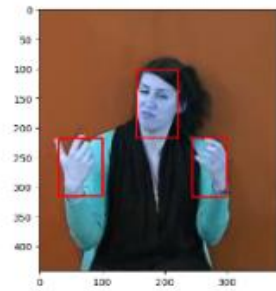


Image: 2
Approximate movement of bounding box: 2.35388896922616 -2.5877321788188773

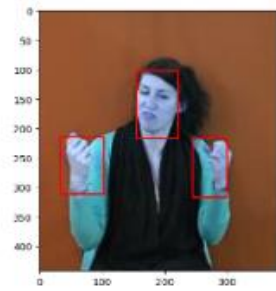


Image: 3
Approximate movement of bounding box: 2.082129805817368 -2.8678153844987637

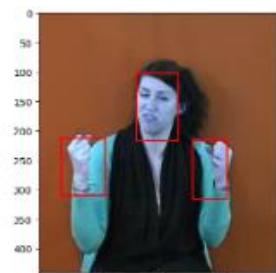


Image: 4
Approximate movement of bounding box: 1.49791884978923 -1.335521338469951

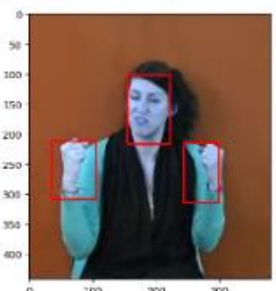


Image: 5
Approximate movement of bounding box: -1.938606281831365 -0.21917188326497796

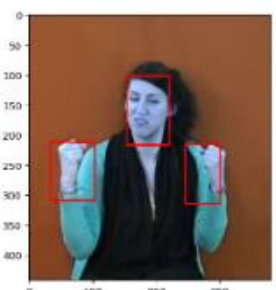


Image: 6
Approximate movement of bounding box: -2.28118682588316 0.6635385354634978

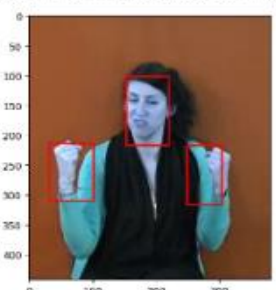


Image: 7
Approximate movement of bounding box: 0.6878239942261941 -0.981292581698492

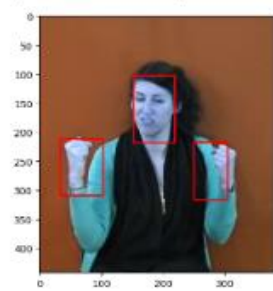


Image: 8
Approximate movement of bounding box: 1.8648736318221698 -0.6196388285824177

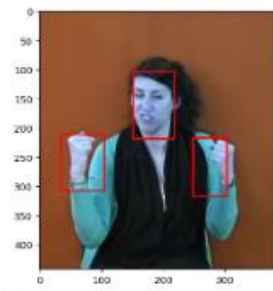


Image: 9
Approximate movement of bounding box: 2.0891948388226976 0.6581142028832119

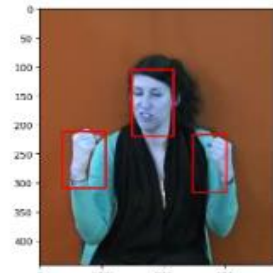


Image: 10
Approximate movement of bounding box: 2.861183674622139 0.5324954058568814



Image: 11
Approximate movement of bounding box: 1.782113387036185 1.2135686435814732

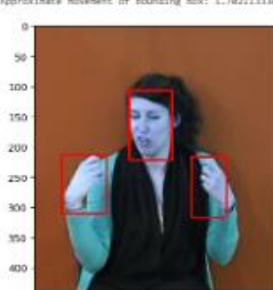
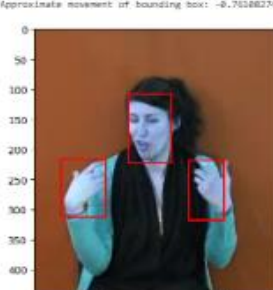


Image: 12
Approximate movement of bounding box: -0.781882743686441 1.9163791625583557



Παρατηρείται ότι στα δύο χέρια, τα οποία αλλάζουν την μορφή τους και μετατοπίζονται πολύ κατά την διάρκεια της κίνησης, υπάρχει δυσκολία για τον Lucas-Kanade να κάνει καλό track. Το bounding box σε αυτές τις περιπτώσεις "χάνει" την κίνηση, και αυτό μάλλον οφείλεται στα διαφορετικά σημεία ενδιαφέροντος που εντοπίζονται μεταξύ των frames της έντονης αλλαγής. Αντίθετα, στο πρόσωπο, που δεν έχει κάποια έντονη και απότομη αλλαγή, κάνουμε πολύ καλό track και στις δύο περιπτώσεις. Επιπρόσθετα, η μέθοδος TV-L1 είναι αποτελεσματικότερη, αφού τα bounding boxes ακολουθούν καλύτερα τα χέρια με μικρότερο σφάλμα.

Στη συνέχεια η υλοποίηση του αλγορίθμου Lucas-Kanade επεκτείνεται για πολλαπλές κλίμακες. Η συνάρτηση **multiscale_lk(I1,I2,rho,epsilon,scales)** αρχικά κάνει downsampling τις αρχικές εικόνες τόσες φορές όσα είναι και τα ζητούμενα scales. Σημειώνεται ότι πριν από κάθε υποδειγματοληψία, μέσω της εντολής *cv2.pyrDown()*, γίνεται βαθυπερατό gaussian φιλτράρισμα για αποφυγή φαινομένων aliasing. Στην συνέχεια, ξεκινάμε από την οπτική ροή των υποδειγματοληπτημένων εικόνων οι οποίες αποτελούν αρχική συνθήκη για την εφαρμογή του μονοκλιμακωτού Lucas-Kanade και τον υπολογισμό των νέων οπτικών ροών. Σημειώνεται ότι πριν την μετάβαση στην επόμενη κλίμακα, οι οπτικές ροές της προηγούμενης γίνονται reshaped στις διαστάσεις της επόμενης κλίμακας και πολλαπλασιάζονται με 2. Η συνάρτηση **ms_tracker(limit=66)** επιστρέφει την θέση των bounding boxes κατά τη διάρκεια της κίνησης εφαρμόζοντας την πολυκλιμακωτή εκδοχή του αλγορίθμου, μερικά αποτελέσματα φαίνονται πιο κάτω για rho=5, epsilon=0.01, scales=5:

Image: 1
Approximate movement of bounding box: 0.25367939283472146 -6.0702483381304955



Image: 3
Approximate movement of bounding box: 5.402244128302822 -0.33047663468055516

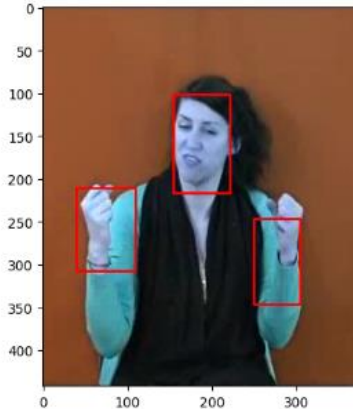


Image: 2
Approximate movement of bounding box: 5.330437594266945 -3.300231247828008



Image: 4
Approximate movement of bounding box: 3.6023354602985243 0.14968337936194268



Image: 5
Approximate movement of bounding box: -2.4127812156146543 0.05822777978545572

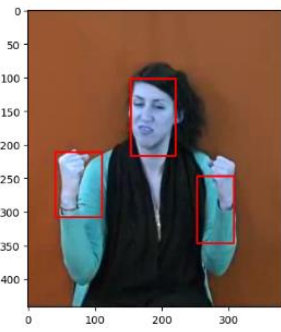


Image: 7
Approximate movement of bounding box: 1.0357756481410456 -1.008363997952164

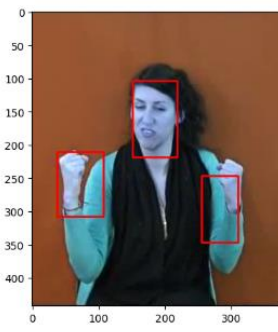


Image: 6
Approximate movement of bounding box: -4.09203353886486 1.0753554595559132

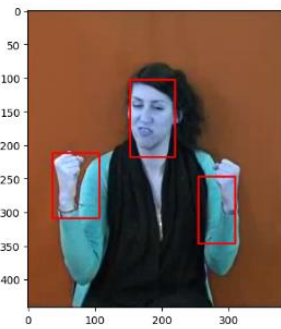


Image: 8
Approximate movement of bounding box: 3.463066442065039 -1.190886177603756

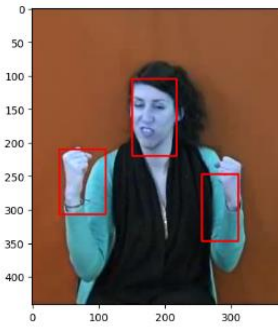


Image: 9
Approximate movement of bounding box: 4.193200450706081 0.856142895076685



Image: 11
Approximate movement of bounding box: 6.587626564266665 5.1434509254111065

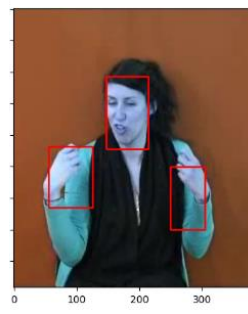
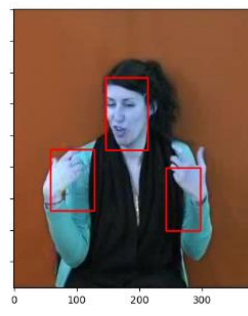


Image: 10
Approximate movement of bounding box: 5.380456270568404 4.588445316295888



Image: 12
Approximate movement of bounding box: 3.8935176440945924 4.467676895636473



Παρατηρείται στον πολυκλιμακωτό Lucas-Kanade πιο αργή σύγκλιση της οπτικής ροής. Παράλληλα, ενώ φαίνονται καλύτερα αποτελέσματα στα bounding boxes για το δεξί χέρι, για το αριστερό χέρι τα αποτελέσματα είναι λίγο χειρότερα. Η πολυκλιμακωτή εκδοχή του αλγόριθμου είναι πιο αξιόπιστη.

Ανάλυση παραμέτρων $\rho_0, e, thr, scales$

Στο τελευταίο στάδιο γίνεται μια ανάλυση των παραμέτρων των συναρτήσεων $lk()$, $displ()$ και $multiscale_lk()$ για τις εξής τιμές:

$\rho_0=[1,2,3,4,5]$

$e=[0.01,0.025,0.05,0.075,1]$

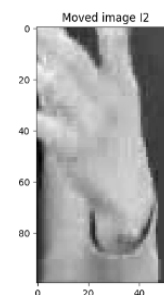
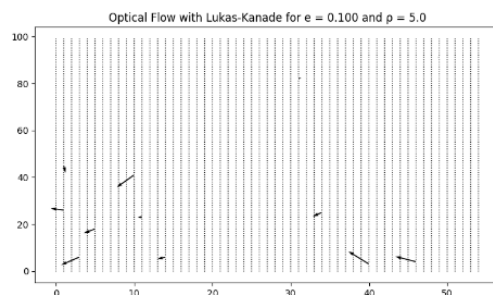
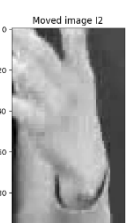
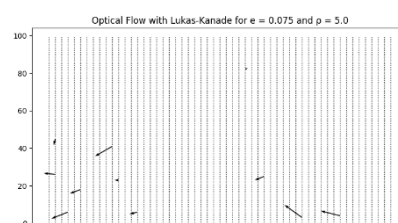
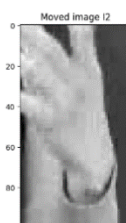
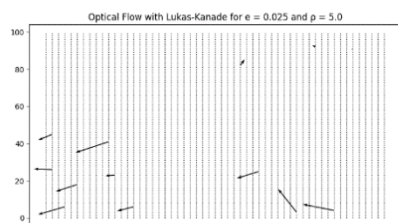
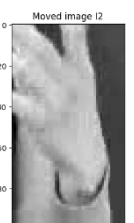
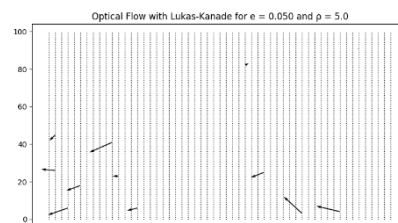
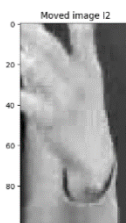
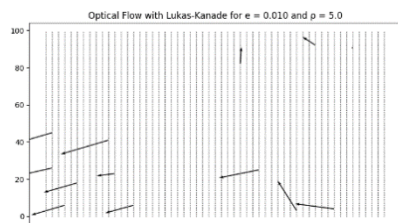
$scales=[1,2,3,4,5]$

$thr=[0.001,0.1,0.2,0.5,0.75]$

Σημειώνεται ότι για λόγους συντομίας η ανάλυση για τις παραμέτρους ρ_0, e θα γίνει στον μονοκλιμακωτό Lucas-Kanade, για την $scales$ στον πολυκλιμακωτό, ενώ για το threshold στην υπορουτίνα $displ()$

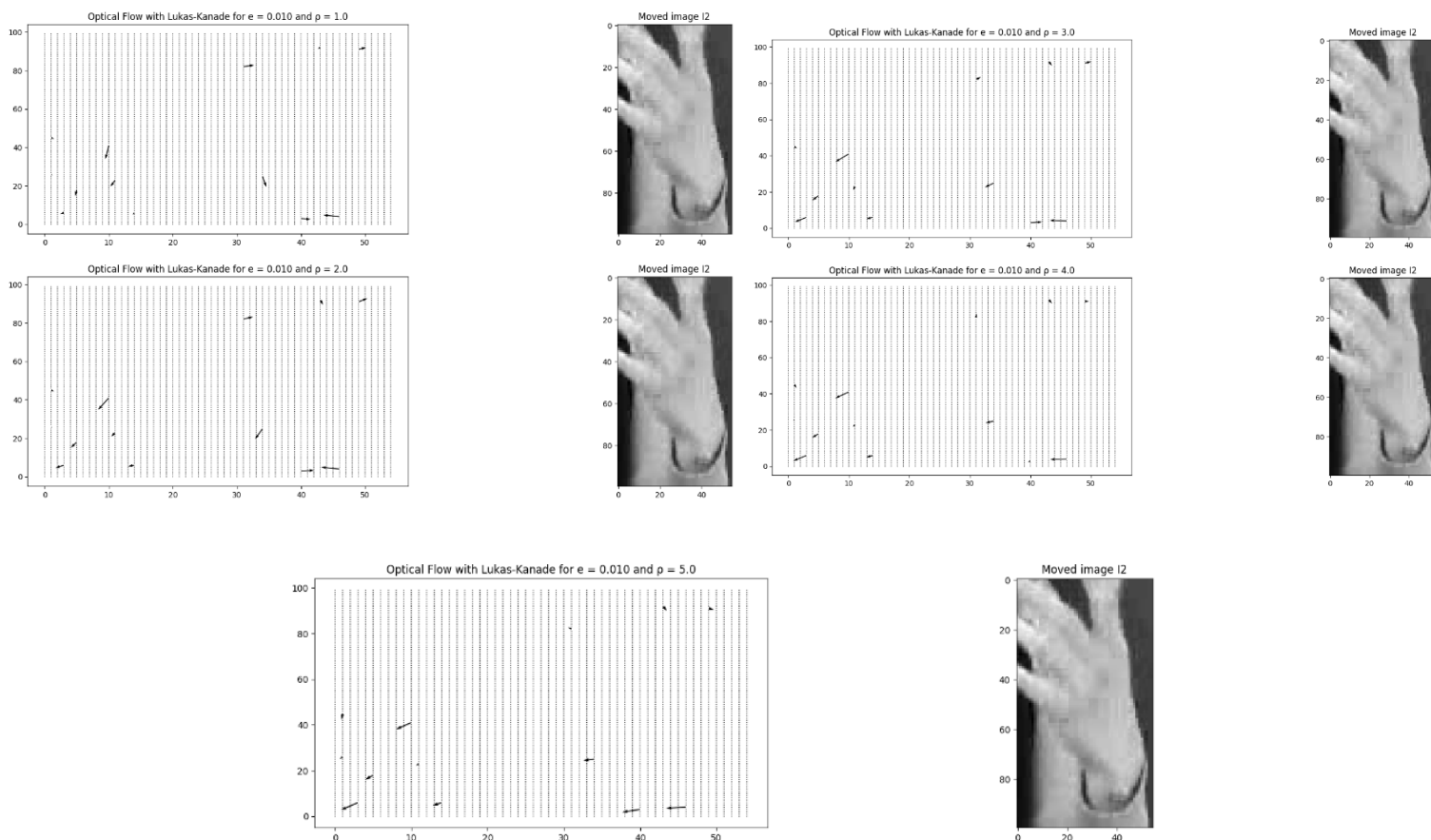
Αποτελέσματα ανάλυσης για την σταθερά κανονικοποίησης e

Η σταθερά ϵ έχει σκοπό να βελτιώσει το αποτέλεσμα σε επίπεδες περιοχές με μειωμένη υφή και άρα μειωμένη πληροφορία για τον υπολογισμό της οπτικής ροής. Άρα, αναμένεται ότι αύξηση της σταθεράς θα οδηγήσει σε πιο “ευσταθή” αποτελέσματα, γεγονός που επαληθεύεται παρακάτω. Η μετατοπισμένη εικόνα έχει $dx=10$, $dy=10$.



Αποτελέσματα ανάλυσης για την παράμετρο ρ_0 της Gaussian συνέλιξης

Η παράμετρος ρ_0 , παρατηρείται ότι, ομοίως με την παράμετρο e , μικρές τιμές της παραμέτρου ρ_0 μειώνουν την αξιοπιστία των αποτελεσμάτων της μεθόδου, μια και προκύπτουν ακμές που δεν αντικατοπτρίζουν την πραγματική μετατόπιση από το ένα frame στο επόμενο. Αντίθετα, μεγαλύτερες τιμές (όπως $\rho_0=4$ ή $\rho_0=5$) οδηγούν σε πολύ πιο ευκρινή αποτελέσματα. Η μετατοπισμένη εικόνα έχει $dx=5$, $dy=5$.



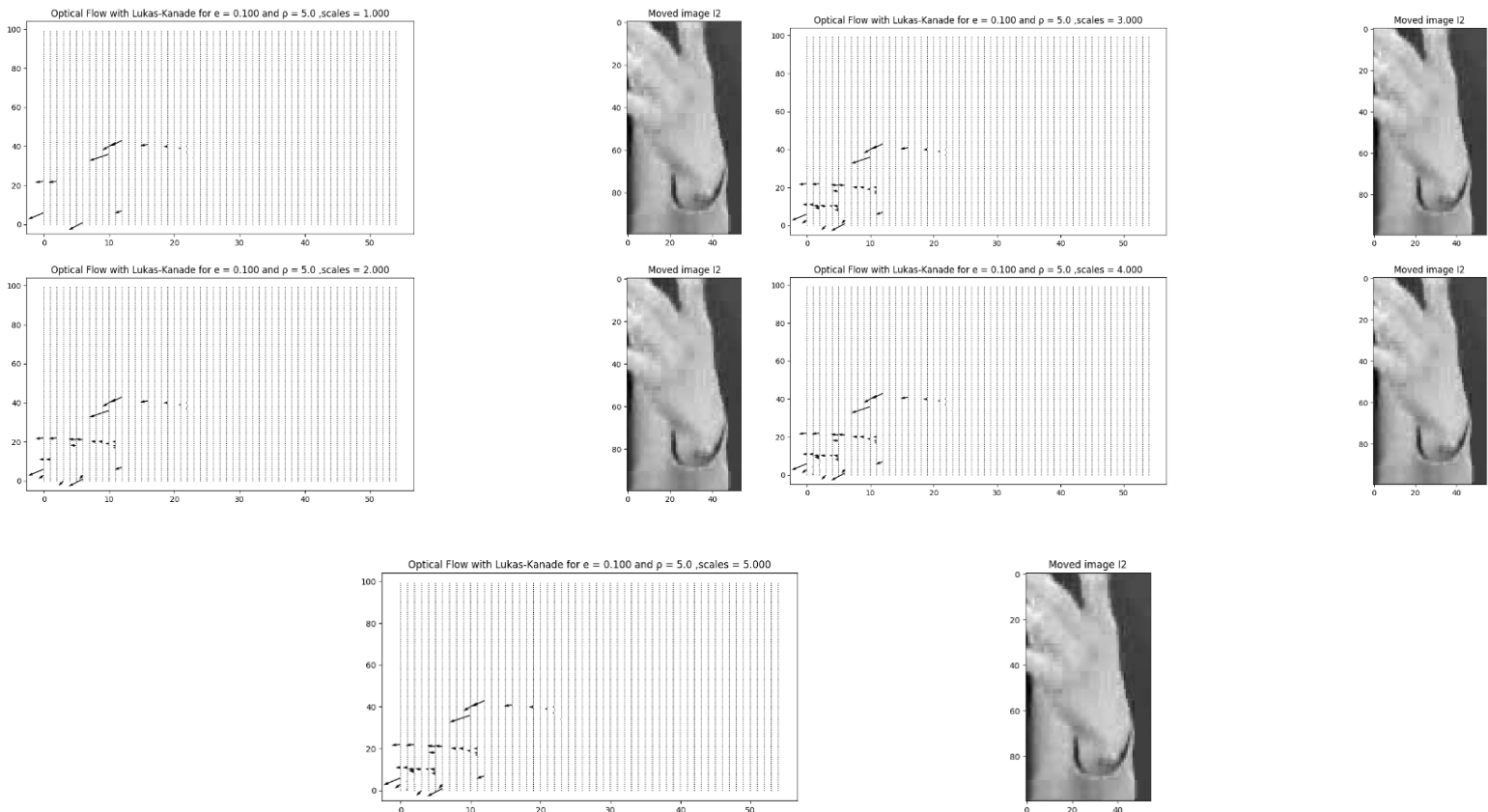
Αποτελέσματα ανάλυσης για την παράμετρο threshold

Παρατηρείται ότι η αύξηση της παραμέτρου κατωφλίου, οδηγεί σε κατ'απόλυτον αύξηση της μέσης μετατόπισης ανά άξονα, επομένως με ένα μεγαλύτερο κατώφλι ανιχνεύονται μεγάλες κινήσεις σε ένα βιντεο, ενώ κρατώντας μικρή τιμή, λαμβάνονται οι μικρότερες μεταβολές σε οπτική ροή.

$e = 0.100$	$\rho = 5.000$	threshold = 0.001	$dx = -1.300$	$dy = -1.338$
$e = 0.100$	$\rho = 5.000$	threshold = 0.100	$dx = -2.119$	$dy = -1.809$
$e = 0.100$	$\rho = 5.000$	threshold = 0.200	$dx = -2.411$	$dy = -1.871$
$e = 0.100$	$\rho = 5.000$	threshold = 0.500	$dx = -3.055$	$dy = -2.233$
$e = 0.100$	$\rho = 5.000$	threshold = 1.000	$dx = -4.284$	$dy = -0.766$

Αποτελέσματα ανάλυσης για την παράμετρο scales

Η αλλαγή κλίμακας επιφέρει αισθητή διαφοροποίηση μεταξύ των αποτελεσμάτων για $scales=1$ και $scales>1$. Ωστόσο, οι διαφορές των διανυσμάτων ροής για $scales>1$ είναι πρακτικά αμελητέες, γεγονός που υποδηλώνει την σύγκλιση του πολυκλιμακωτού Lukas-Kanade για $scales=2$. Η μετατοπισμένη εικόνα έχει $dx=10$, $dy=10$.



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

Στο μέρος 2 του εργαστηριακού project θα γίνει κατηγοριοποίηση βίντεο σε 3 κατηγορίες ανθρωπίνων δράσεων: σε τρέξιμο, περπάτημα και μποξ. Για να γίνει αυτό αρχικά εξάγονται χωρο-χρονικά χαρακτηριστικά από τα βίντεο. 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 (π.χ. bag of visual words) ικανή να αναπαραστήσει τη στατιστική κατανομή τους και να προχωρήσει στα επόμενα στάδια της αναγνώρισης.

2.1 Χωρο-χρονικά Σημεία Ενδιαφέροντος

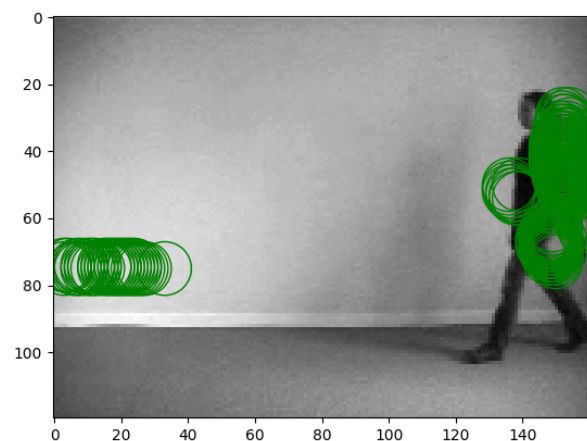
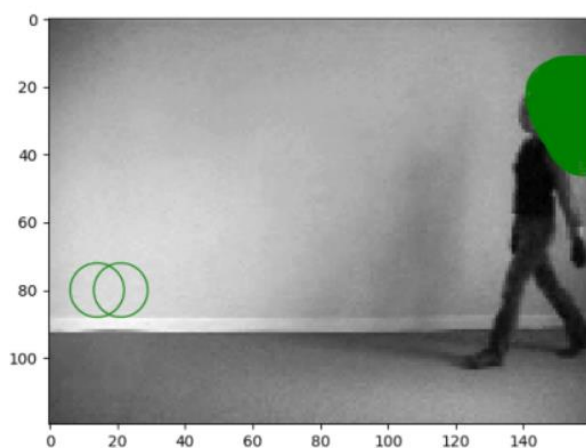
Στην παρούσα άσκηση υλοποιούνται δύο ανιχνευτές: 1) Harris detector και 2) Gabor detector.

Ο ανιχνευτής Harris αποτελεί μία προέκταση σε 3 διαστάσεις του ανιχνευτή γωνιών Harris-Stephens που υλοποιήσαμε στην 1η εργαστηριακή άσκηση προσθέτοντας έναν 2Δ τανυστή και τη χρονική παράγωγο. Η συνάρτηση **harrisDetector3D(video,sigma,t,scale,k)** δέχεται ως είσοδο τα frames από το εκάστοτε βίντεο, τη χωρική και χρονική σταθερά sigma ,t , την κλίμακα και τη σταθερά k και επιστρέφει τα 600 πρώτα keypoints για το αντίστοιχο frame. Το κριτήριο γωνιότητας που εφαρμόστηκε είναι: $H(x,y,t)=\det(M(x,y,t))-k \cdot \text{trace}^3(M(x,y,t))$ και εφαρμόζεται ένα έξτρα κριτήριο για απόρριψη σημείων που αντιστοιχούν σε σχετικά ομαλές περιοχές (κατώφλι= 0.25).

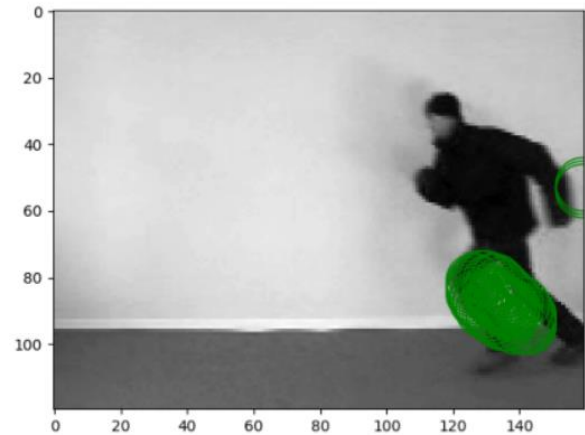
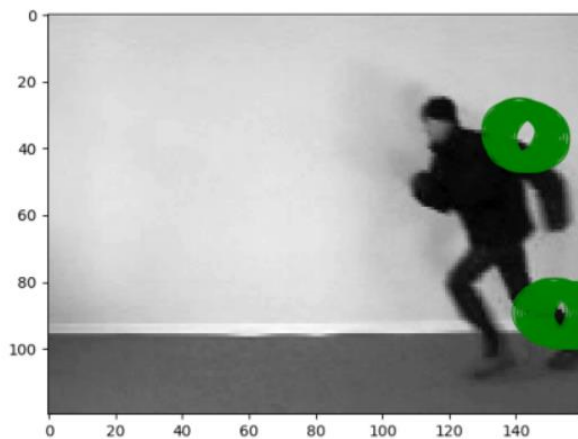
Ο ανιχνευτής Gabor βασίζεται στο χρονικό φιλτράρισμα του βίντεο με ένα ζεύγος Gabor φίλτρων αφού πρώτα έχει υποστεί εξομάλυνση στις χωρικές διαστάσεις μέσω ενός 2Δ γκαουσιανού πυρήνα $g(x,y;\sigma)$. Η συνάρτηση **Gabor(video,sigma,t)** δέχεται ως είσοδο τα frames από το εκάστοτε βίντεο, τη χωρική και χρονική σταθερά sigma ,t και επιστρέφει τα πρώτα 600 keypoints που ανιχνεύει για το εκάστοτε frame. Το κριτήριο γωνιότητας που εφαρμόστηκε είναι : $H(x, y, t) = (I(x, y, t) * g * hev)^2 + (I(x, y, t) * g * hod)^2$ και εφαρμόζεται ένα έξτρα κριτήριο για απόρριψη σημείων που αντιστοιχούν σε σχετικά ομαλές περιοχές (κατώφλι= 0.25).

Για την απεικόνιση χρησιμοποιήθηκε η δοσμένη συνάρτηση **show_detection**. Ακολουθούν αντιπροσωπευτικά frames με τα αποτελέσματα των δυο ανιχνευτών για κάθε μια από τις τρεις κλάσεις των βίντεο. Στους φακέλους HarrisDetector και Gabor βρίσκονται τα frames για ένα βίντεο απο κάθε κλάση με τα σημεία που ανίχνευσε ο αντίστοιχος αλγόριθμος. Τα βίντεο στο σημείο αυτό έχουν διασπαστεί σε 20 frames το καθένα για απλοποίηση και οι παραμέτροι είναι sigma=4 , t=1.5, scale= 2 , k=0.005.

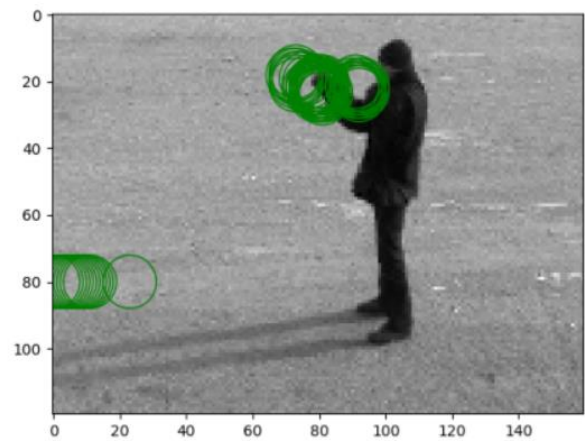
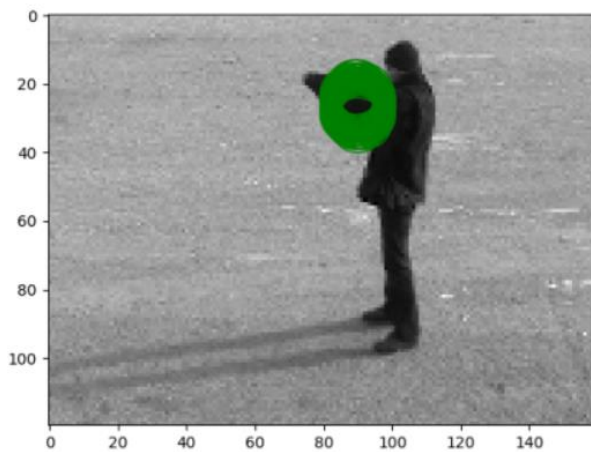
Για walking , harrisDetector(αριστερά) και Gabor(δεξιά) αντίστοιχα:



Για *running* , *harrisDetector*(αριστερά) και *Gabor*(δεξιά) αντίστοιχα:



Για *boxing* , *harrisDetector*(αριστερά) και *Gabor*(δεξιά) αντίστοιχα:



Παρατηρώντας όλα τα εξαγόμενα frames με τα εντοπισμένα σημεία ενδιαφέροντος φαίνεται πως ο ανιχνευτής Harris εντοπίζει σχεδόν κάθε φορά την κίνηση στα βίντεο ενώ ο ανιχνευτής Gabor εντοπίζει ένα μικρό μέρος των κινήσεων. Ωστόσο ο Gabor ανιχνευτής είναι πιο εύστοχος από τον Harris. Αυτό συμβαίνει καθώς ο ανιχνευτής Harris εντοπίζει κυρίως γωνίες στα frame των βίντεο όπως φαίνεται και στα παραπάνω παραδειγματικά frames. Αντίθετα στον Gabor ανιχνευτή τα σημεία που εντοπίζονται είναι πιο στοχευμένα και αυτό φαίνεται και από τα παραπάνω παραδείγματα. Κατά το τρέξιμο εστιάζονται οι γωνίες στα πόδια, στο περπάτημα εστιάζονται χαρακτηριστικά στον κορμό και στα πόδια, ενώ στο μποξ εντοπίζονται οι γωνίες στον αγκώνα, καρπό και παλάμη. Αξίζει να σημειωθεί, ότι ο ανιχνευτής Gabor εντοπίζει περισσότερα σημεία που δεν υπάρχει πραγματική κίνηση στο background, το οποίο μπορεί να μειωθεί μεταβάλλοντας την τιμή κατωφλίου απόρριψης σημείων.

2.2 Χωρο-χρονικοί Ιστογραφικοί Περιγραφητές

Σε αυτό το βήμα εξάγονται οι χωρο-χρονικοί περιγραφητές HOG, HOF και ένας συνδυασμός HOG και HOF. Για γίνει αυτό, μετά τον υπολογισμό της παραγώγου και της οπτικής ροής, εξάγουμε τετραγωνικές περιοχές πλευράς σ (η σ που χρησιμοποιήθηκε για την εξαγωγή σημείων ενδιαφέροντος) τόσο για τον x άξονα όσο και για τον y , στο οποίο θα υπολογίσουμε τους ιστογραφικούς περιγραφητές. Οι περιγραφητές υπολογίζονται με την δοθείσα συνάρτηση `orientation_histogram` η οποία δέχεται ως είσοδο τα G_x , G_y που ορίζουν ένα διανυσματικό πεδίο. Για τον HOG τα G_x , G_y υπολογίζονται από το διάνυσμα κλίσης αλλά για τον HOF απαιτείται ένα πυκνό πεδίο οπτικής ροής και για αυτό παίρνουμε τα G_x , G_y από την TLV1 οπτική ροή. Ο περιγραφητής συνδυασμού των HOG/HOF προκύπτει με συνένωση των δύο. Η συνάρτηση `hist_desc(video, feats, desc_type, nbins)` δέχεται ως είσοδο τα frames του βίντεο, τα σημεία ενδιαφέροντος από τους ανιχνευτές του προηγούμενου βήματος, το είδος του περιγραφητή ("HOG", "HOF" or "HOG/HOF") και τον αριθμό των bins του ιστογράμματος και επιστρέφει τον επιθυμητό ιστογραφικό περιγραφητή για το εκάστοτε βίντεο.

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

Στη συνέχεια γίνεται η κατηγοριοποίηση των βίντεο σε κατηγορίες με χρήση BoVW αναπαραστάσεων βασισμένων στα HOG/HOF χαρακτηριστικά. Αρχικά γίνεται διαχωρισμός των αρχείων σε δύο σύνολα (εκπαίδευσης και δοκιμής) με αντίστοιχα labels. Για την τελική αναπαράσταση χρησιμοποιήθηκε η συνάρτηση που μας δόθηκε `bag_of_words`. Αφού έχουμε εξάγει αναπαραστήσεις για όλα τα βίντεο εκπαιδεύουμε το μοντέλο χρησιμοποιώντας την συνάρτηση `svm_train_test`. Ακολουθούν για κάθε συνδυασμό ανιχνευτών/περιγραφητών η ακρίβεια και αναλυτικοί πίνακες που δείχνουν την κατηγοριοποίηση των δράσεων. Χρησιμοποιήθηκαν `n_bins=9` και `num_centers=400`.

Detector: Harris
Descriptor: HOG

The accuracy for the combination is 75.000%

Results:

Video Name	Correct	Prediction
person07_walking_d4_uncomp.avi	walk	walk True
person08_walking_d2_uncomp.avi	walk	walk True
person11_walking_d1_uncomp.avi	walk	walk True
person20_walking_d3_uncomp.avi	walk	run False
person06_boxing_d1_uncomp.avi	box	box True
person07_boxing_d3_uncomp.avi	box	box True
person11_boxing_d2_uncomp.avi	box	box True
person15_boxing_d4_uncomp.avi	box	walk False
person10_running_d2_uncomp.avi	run	run True
person20_running_d1_uncomp.avi	run	run True
person24_running_d3_uncomp.avi	run	run True
person25_running_d4_uncomp.avi	run	walk False

Detector: Harris
Descriptor: HOF

The accuracy for the combination is 33.333%

Results:

Video Name	Correct	Prediction
person07_walking_d4_uncomp.avi	walk	walk True
person08_walking_d2_uncomp.avi	walk	run False
person11_walking_d1_uncomp.avi	walk	walk True
person20_walking_d3_uncomp.avi	walk	walk True
person06_boxing_d1_uncomp.avi	box	walk False
person07_boxing_d3_uncomp.avi	box	walk False
person11_boxing_d2_uncomp.avi	box	walk False
person15_boxing_d4_uncomp.avi	box	walk False
person10_running_d2_uncomp.avi	run	walk False
person20_running_d1_uncomp.avi	run	walk False
person24_running_d3_uncomp.avi	run	walk False
person25_running_d4_uncomp.avi	run	run True

Detector: Harris
Descriptor: HOG_HOF

The accuracy for the combination is 50.000%

Results:

Video Name	Correct	Prediction
person07_walking_d4_uncomp.avi	walk	walk True
person08_walking_d2_uncomp.avi	walk	walk True
person11_walking_d1_uncomp.avi	walk	walk True
person20_walking_d3_uncomp.avi	walk	walk True
person06_boxing_d1_uncomp.avi	box	walk False
person07_boxing_d3_uncomp.avi	box	walk False
person11_boxing_d2_uncomp.avi	box	walk False
person15_boxing_d4_uncomp.avi	box	walk False
person10_running_d2_uncomp.avi	run	run True
person20_running_d1_uncomp.avi	run	walk False
person24_running_d3_uncomp.avi	run	run True
person25_running_d4_uncomp.avi	run	walk False

Detector: Gabor
Descriptor: HOG

The accuracy for the combination is 58.333%

Results:

Video Name	Correct	Prediction	
person07_walking_d4_uncomp.avi	walk	run	False
person08_walking_d2_uncomp.avi	walk	walk	True
person11_walking_d1_uncomp.avi	walk	walk	True
person20_walking_d3_uncomp.avi	walk	run	False
person06_boxing_d1_uncomp.avi	box	box	True
person07_boxing_d3_uncomp.avi	box	box	True
person11_boxing_d2_uncomp.avi	box	box	True
person15_boxing_d4_uncomp.avi	box	walk	False
person10_running_d2_uncomp.avi	run	run	True
person20_running_d1_uncomp.avi	run	box	False
person24_running_d3_uncomp.avi	run	run	True
person25_running_d4_uncomp.avi	run	walk	False

Detector: Gabor
Descriptor: HOG_HOF

The accuracy for the combination is 33.333%

Results:

Video Name	Correct	Prediction	
person07_walking_d4_uncomp.avi	walk	run	False
person08_walking_d2_uncomp.avi	walk	walk	True
person11_walking_d1_uncomp.avi	walk	walk	True
person20_walking_d3_uncomp.avi	walk	walk	True
person06_boxing_d1_uncomp.avi	box	walk	False
person07_boxing_d3_uncomp.avi	box	walk	False
person11_boxing_d2_uncomp.avi	box	walk	False
person15_boxing_d4_uncomp.avi	box	run	False
person10_running_d2_uncomp.avi	run	walk	False
person20_running_d1_uncomp.avi	run	walk	False
person24_running_d3_uncomp.avi	run	walk	False
person25_running_d4_uncomp.avi	run	run	True

Detector: Gabor
Descriptor: HOF

The accuracy for the combination is 33.333%

Results:

Video Name	Correct	Prediction	
person07_walking_d4_uncomp.avi	walk	run	False
person08_walking_d2_uncomp.avi	walk	walk	True
person11_walking_d1_uncomp.avi	walk	walk	True
person20_walking_d3_uncomp.avi	walk	walk	True
person06_boxing_d1_uncomp.avi	box	walk	False
person07_boxing_d3_uncomp.avi	box	walk	False
person11_boxing_d2_uncomp.avi	box	walk	False
person15_boxing_d4_uncomp.avi	box	run	False
person10_running_d2_uncomp.avi	run	walk	False
person20_running_d1_uncomp.avi	run	walk	False
person24_running_d3_uncomp.avi	run	walk	False
person25_running_d4_uncomp.avi	run	run	True

Παρατηρείται ότι γενικά οι περιγραφητές δεν πετυχαίνουν ιδιαίτερα υψηλά ποσοστά. Συγκεκριμένα ο περιγραφής HOF έχει την ίδια επίδοση και με τους δύο ανιχνευτές και αναγνωρίζει σχεδόν όλα τα βίντεο ως walk. Αυτό το πρόβλημα ίσως λυθεί αν μειωθεί ο αριθμός βίντεο με label walk στο training για εκπαιδευτεί το μοντέλο περισσότερο στην αναγνώριση του τρεξίματος και του μπόξ. Ο καλύτερος συνδυασμός επιτυγχάνεται με τον ανιχνευτή HarrisDetector και περιγραφητή HOG με επίδοση 75%. Γενικά, στις περισσότερες λανθασμένες ταξινομήσεις συγχέεται το running με το walking, το οποίο είναι λογικό, καθώς το running αποτελεί επιτάχυνση του walking και οι κινήσεις είναι πολύ παρόμοιες. Η επίδοση των μοντέλων θα μπορούσε να αυξηθεί, με μια βελτιστοποίηση των παραμέτρων ανιχνευτή/περιγραφητή, αφού για τα πιο πάνω αποτελέσματα χρησιμοποιήθηκαν οι ενδεικτικές τιμές.

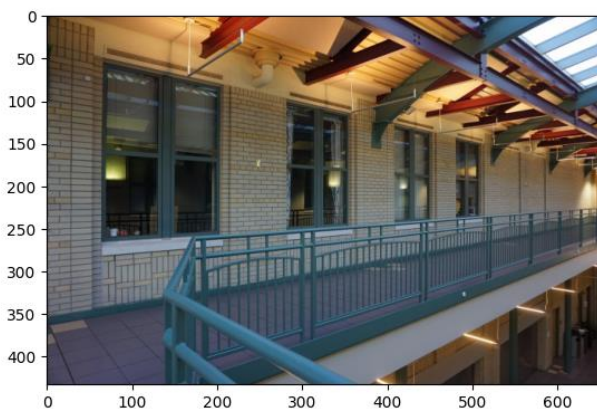
ΜΕΡΟΣ 3: Συνένωση Εικόνων(Image Stitching) για Δημιουργία Πανοράματος

Σκοπός του μέρους αυτού είναι η δημιουργία μιας συνάρτησης για την συνένωση διαδοχικών εικόνων που έχουν ληφθεί από μια κάμερα σε σταθερό σημείο, η οποία έχει περιστραφεί φορές. Η συνάρτηση βασίζεται στον υπολογισμό ενός πίνακα ομογραφίας Η μεταξύ διαδοχικών εικόνων, για την δημιουργία ενός κοινού πλαισίου αναφοράς. Τελικό αποτέλεσμα της διαδικασίας είναι μια εικόνα πανοράματος, με στόχο την τεχνητή αύξηση του πεδίου ορατότητας (field of view).

Η βασική ρουτίνα που χρησιμοποιήθηκε για τον σχηματισμό του ζητούμενου πανοράματος είναι η `stichImages(img1, img2)`, η οποία θα συνενώνει την εικόνα I1 στο πλαίσιο αναφορά I2, με βάση τα επόμενα βήματα που θα ακολουθήσουν. Παρακάτω ακολουθεί η ανάλυση της ρουτίνας για τις δύο πρώτες εικόνες του φακέλου Part 3, ενώ στην συνέχεια θα αποδοθούν περιληπτικά τα αποτελέσματά της και για τις υπόλοιπες.

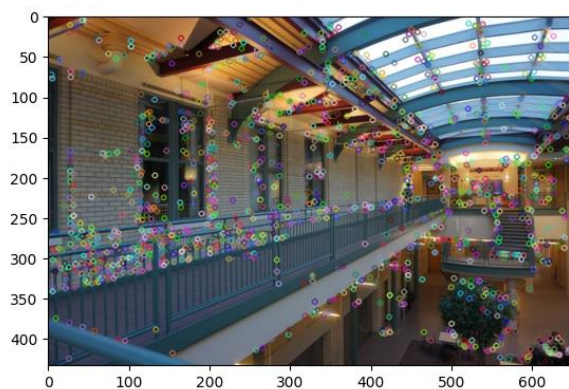
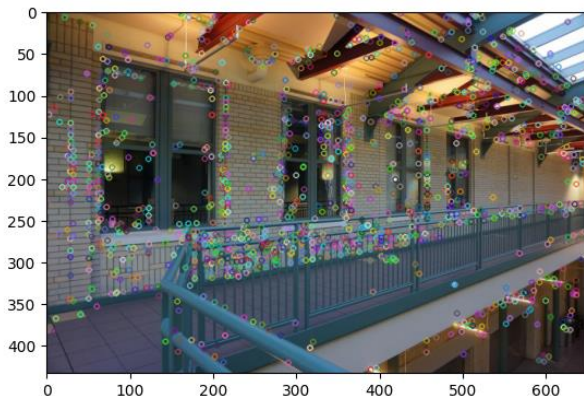
Βήμα 0: Διάβασμα Εικόνων

Αρχικά διαβάζονται οι δύο πρώτες εικόνες μέσω της εντολής `imread()` και απεικονίζονται παρακάτω στο χώρο RGB.



Βήμα 1: Εντοπισμός χαρακτηριστικών ενδιαφέροντος και εξαγωγή περιγραφητών

Εξάγονται τα σημεία ενδιαφέροντος από τις δύο εικόνες, με την χρήση της `SIFT_create()` και απεικονίζονται με τη χρήση της εντολής `drawKeypoints()`. Τα σημεία που εντοπίζονται για τις 2 πρώτες εικόνες είναι τα εξής:



Βήμα 2-3: Διαδικασία ταιριάσματος χαρακτηριστικών των εικόνων και εφαρμογή του κριτηρίου Lowe

Αντιστοιχίζεται κάθε χαρακτηριστικό της εικόνας 1 με τα 2 κοντινότερα χαρακτηριστικά της εικόνας 2, μέσω υπολογισμού της απόστασης των αντίστοιχων περιγραφητών. Μεταξύ των 2 δοθέντων επιλογών, επιλέχθηκε η εύρεση των προσεγγιστικών κοντινότερων γειτόνων, μέσω των ρουτίνων *FlannBasedMatcher*, *knnMatch*. Από τα αποτελέσματα του βήματος 2, παρατηρείται ένα μεγάλο πλήθος αντιστοιχίσεων χαρακτηριστικών ανάμεσα στις 2 εικόνες, αρκετές εκ των οποίων είναι λάθος. Για να γίνει καλό matching θα πρέπει να επιλεγούν τα κοντινότερα σημεία, γ'αυτό και εφαρμόζεται το κριτήριο Lowe με κατώφλι απόστασης ίσο με 0.75, κρατώντας τις αντιστοιχίσεις των οποίων ο λόγος είναι μικρότερος από αυτό. Η διαδικασία αυτή υλοποιείται με τη συνάρτηση **FLANN_matching(img1,img2)** και το αποτέλεσμα φαίνεται πιο κάτω:

Features before Lowe Threshold: 1743
Features after Lowe Threshold: 539



Παρατηρείται αισθητή μείωση του πλήθους αντιστοιχίσεων (σχεδόν 70%) και παράλληλα γίνονται πιο ευδιάκριτες οι περιοχές των 2 εικόνων μεταξύ των οποίων θα πραγματοποιηθεί το stitching.

Βήμα 4: Υπολογισμός ομογραφίας με RANSAC

Σε αυτό το βήμα, με την βοήθεια της RANSAC, υπολογίζεται ο πίνακας ομογραφίας H , ο οποίος είναι χρήσιμος, γιατί μας δείχνει πως απεικονίζονται τα σημεία της 1ης εικόνας στον σύστημα συντεταγμένων της 2ης. Συγκεκριμένα ο πίνακας H είναι:

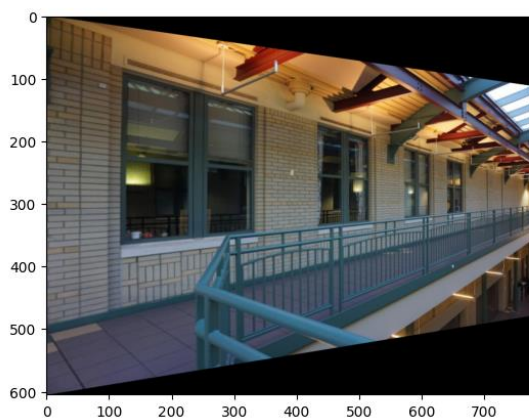
```
[ [ 1.64564307e+00  2.48498767e-03 -3.21468967e+02 ]  
  [ 2.06517328e-01  1.39037577e+00 -7.84873943e+01 ]  
  [ 1.00125163e-03 -2.08206424e-05  1.00000000e+00 ] ]
```


Βήμα 5: Υπολογισμός μετασχηματισμένης εικόνας μέσω *inverse warping*

Η πρώτη εικόνα μετασχηματίζεται σύμφωνα με τον πίνακα ομογραφίας H , ώστε να απεικονιστεί στο χώρο της 2ης εικόνας. Αρχικά υπολογίζονται τα 4 νέα ακριανά σημεία της εικόνας (με την βοήθεια της H) και έπειτα υπολογίζεται ο μετασχηματισμός M , που θα μεταφέρει την 1η εικόνα από τα αρχικά ακριανά σημεία στα νέα. Ο νέος μετασχηματισμός M θα πρέπει από την μία να λάβει υπόψη τον πίνακα ομογραφίας αλλά και την μετατόπιση που μπορεί να πρέπει να εφαρμοστεί ώστε η μετασχηματισμένη 1η εικόνα να «ευθυγραμμιστεί» με την 2η. Τελικά ορίζεται ο πίνακας μετατόπισης

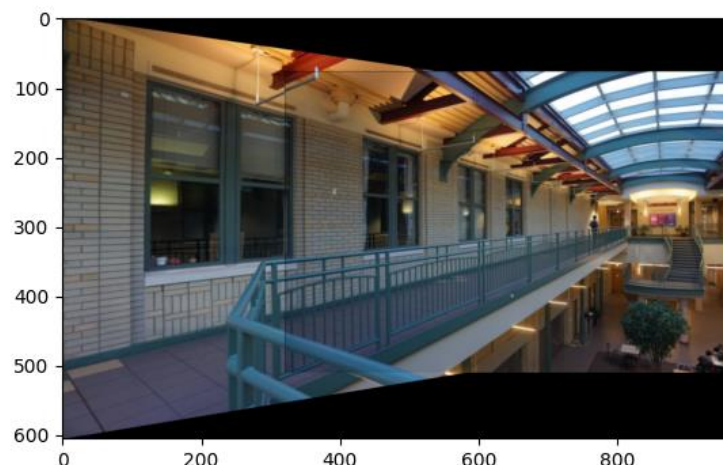
$$T = \begin{bmatrix} 1 & 0 & -A[0] \\ 0 & 1 & -A[1] \\ 0 & 0 & 1 \end{bmatrix}$$

όπου $A[0], A[1]$ είναι οι (x, y) συντεταγμένες του πάνω αριστερά γωνιακού σημείου της 1ης εικόνας. Συνεπώς, ο μετασχηματισμός M είναι $M = T * H$ (ώστε να εφαρμόσουμε και τους 2 μετασχηματισμούς). Η συνάρτηση **projectionImage(H, img)** δέχεται ως είσοδο την εικόνα και τον πίνακα ομογραφίας και επιστρέφει την μετασχηματισμένη εικόνα, το αποτέλεσμα φαίνεται παρακάτω.



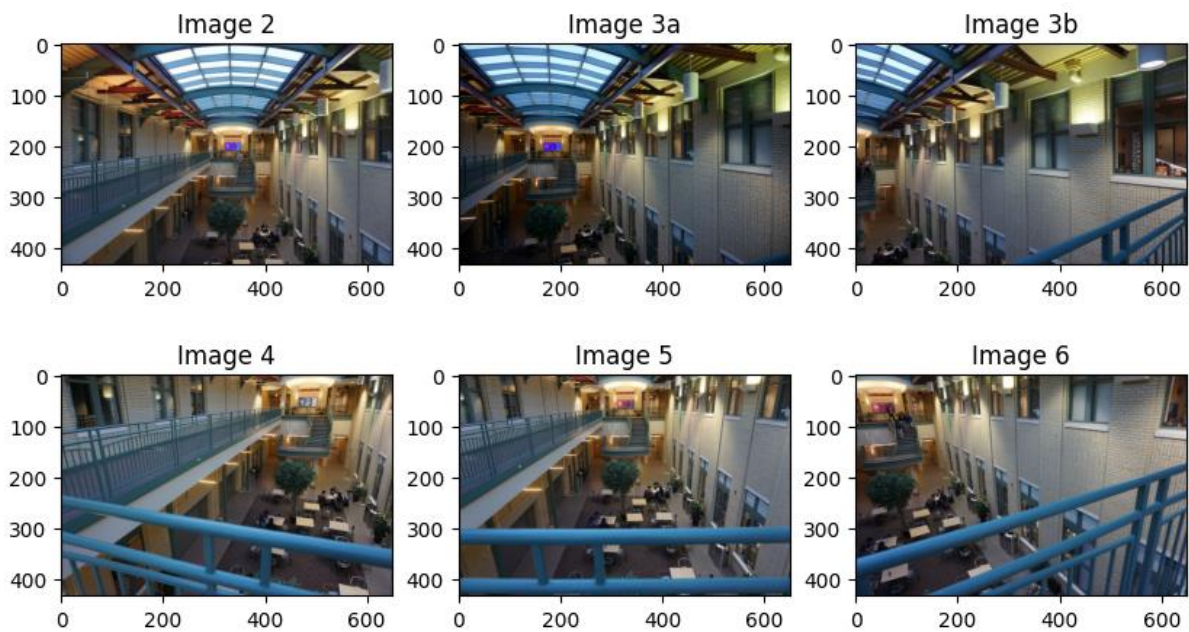
Βήμα 6: Συνένωση των εικόνων

Τέλος γίνεται η συνένωση των δύο εικόνων, με την βοήθεια των προηγούμενων βημάτων. Υπολογίζεται το μέγεθος της εικόνας που θα παραχθεί από την συνένωση, μεταφέρονται οι εικόνες στο νέο μέγεθος, και γίνεται η ένωση (μηδενίζοντας τα pixels της μίας εκεί που υπάρχει επικάλυψη). Η συνάρτηση **mergeWarpedImages($img1_warped, img1b, img1_topleft_coords$)** δέχεται ως είσοδο την warped εικόνα, την εικόνα με την οποία θα γίνει συνένωση και τις συντεταγμένες του πάνω αριστερά σημείου της δεύτερης εικόνας και επιστρέφει το εξής αποτέλεσμα:

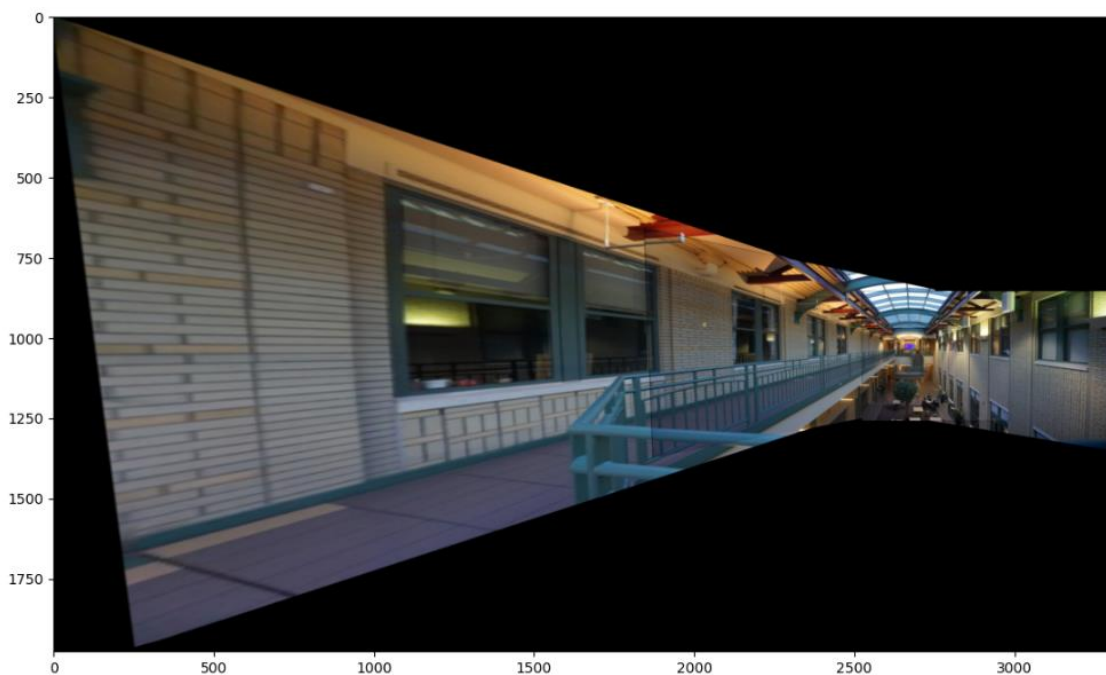


Τελική συνένωση των εικόνων-Πανόραμα

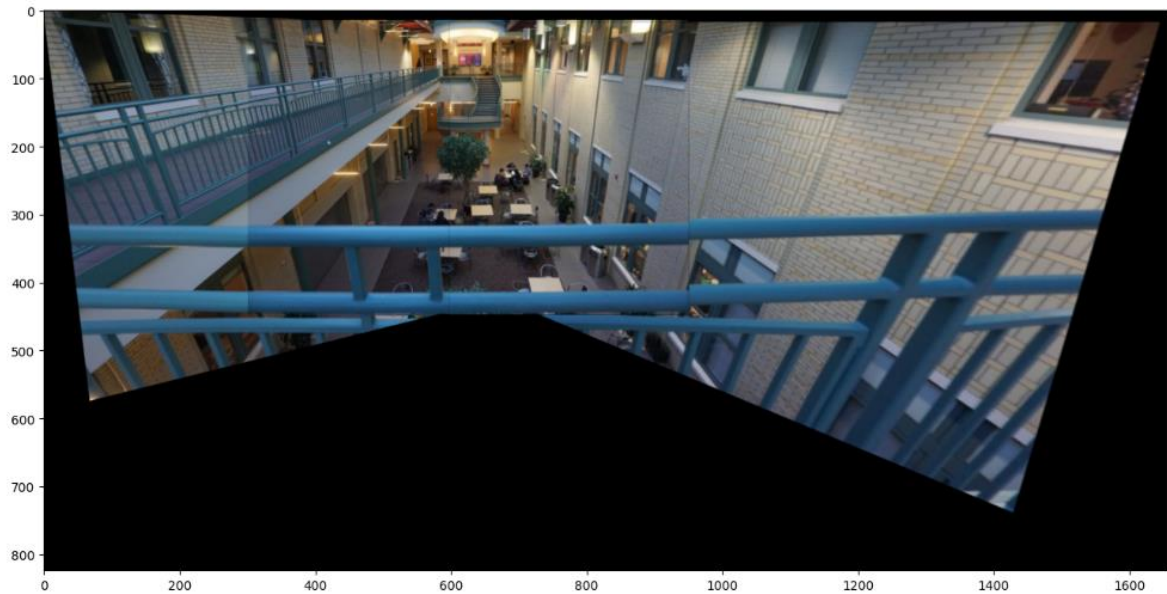
Αρχικά παρτίθενται οι υπόλοιπες εικόνες που ζητείται να γίνει συνένωση.



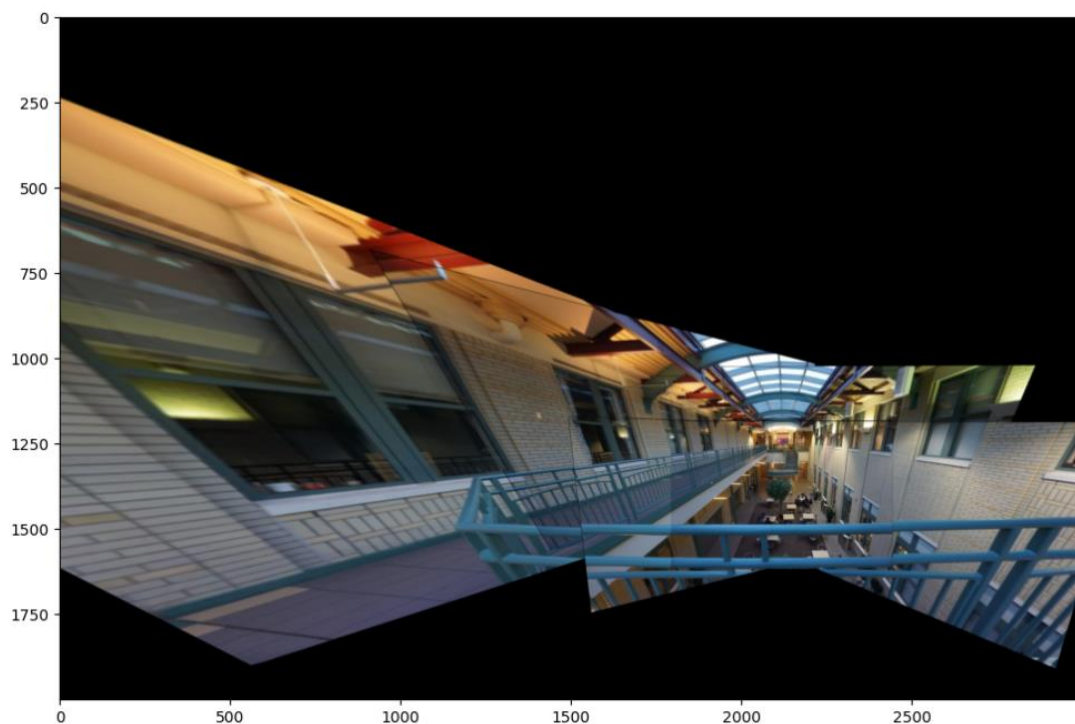
Στη συνέχεια γίνεται επαναληπτική κλήση της συνάρτησης **stichImages()**, για τη δημιουργία του πανοράματος. Είναι σημαντική η σειρά με την οποία θα γίνει η συνένωση για να υπάρξει καλό αποτέλεσμα. Πρώτα γίνεται συνένωση των εικόνων 2 και 3α και μετά συνένωση με την πρώτη εικόνα που δημιουργήθηκε. Το αποτέλεσμα είναι το εξής:



Κατόπιν, γίνεται συνένωση των εικόνων 4, 5, 6 με τη ακολουθία: 4->5, 6->5 και 45->65. Ο λόγος αυτής τη ακολουθίας είναι ότι και στις δύο παραγόμενες εικόνες έχουν την εικόνα 5 ως «κοινή» και μη παραμορφωμένη. Έτσι, όταν εφαρμοστεί η παραπάνω διαδικασία, οι δύο εικόνες θα έχουν κοινά σημεία τα σημεία της 5, γεγονός που φαίνεται στην πιο κάτω εικόνα.



Τέλος η συνένωση των δύο τελευταίων εικόνων μας δίνει το τελικό πανόραμα.



Παρατηρείται ότι το αποτέλεσμα είναι αρκετά ικανοποιητικό. Μια βελτίωση του αποτελέσματος θα ήταν και η μεταβολή της φωτεινότητας του τελικού αποτελέσματος, ώστε να μην υπάρχουν έντονες διαφορές μεταξύ των εικόνων.