



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών

Ροή Σ: Όραση Υπολογιστών (8^ο Εξάμηνο)

Εργαστηριακή Άσκηση 2

Δημήτρης Μπακάλης 03118163

Μέρος 1

1.1 Ανίχνευση Δέρματος Προσώπου και Χεριών:

Στο πρώτο υπομέρος του μέρους της εργαστηριακής άσκησης, που αφορά την εκτίμηση οπτικής ροής, θα επιχειρήσουμε να εξάγουμε 3 πλαίσια αρχικοποίησης (για πρόσωπο και τα δύο χέρια), στα οποία θα εστιάσουμε την αναζήτηση της οπτικής ροής.

Η διαδικασία, αυτή, θα υλοποιηθεί μέσω της αναγνώρισης δέρματος, μοντελοποιώντας το χρώμα του με μία δισδιάστατη γκαουσιανή κατανομή. Οι δύο διαστάσεις της κατανομής αφορούν τις συνιστώσες Cb, Cr του χρωματικού χώρου YCbCr (το luminance αγνοείται, καθώς είναι ενδεικτικό της φωτεινότητας της εικόνας). Ο λόγος, που επιλέχθηκε ο παραπάνω χώρος είναι πως, αν θεωρήσουμε ένα classification πρόβλημα για το αν ένα χρώμα είναι από δέρμα ή όχι και επιλέξουμε για χαρακτηριστικά των δεδομένων μας τις συνιστώσες Cb, Cr, το πρόβλημα γίνεται σχεδόν γραμμικά διαχωρίσιμο.

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

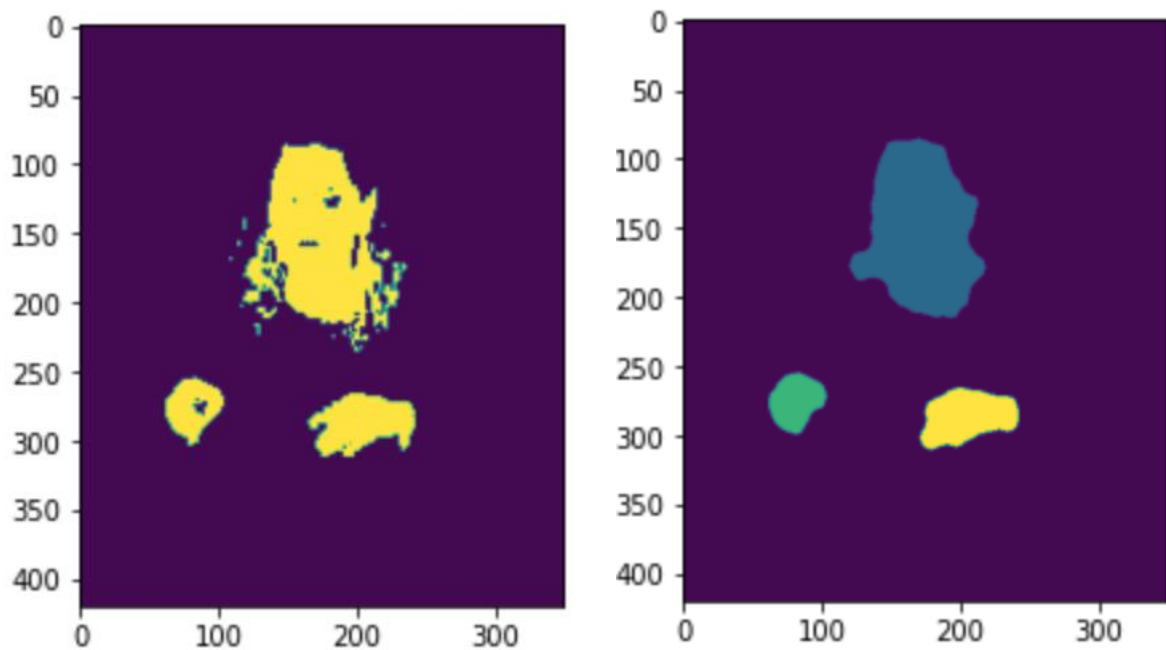
Για τις παραμέτρους μ , Σ της γκαουσιανής, με maximum likelihood, μπορεί να αποδειχθεί, το διαισθητικά προφανές, ότι οι τιμές που ελαχιστοποιούν το σφάλμα είναι οι ακόλουθες: $\mu = [\mu_{Cb}, \mu_{Cr}]$ και

$$\Sigma = \begin{bmatrix} \sigma_{Cb}^2 & \sigma_{CbCr} \\ \sigma_{CrCb} & \sigma_{Cr}^2 \end{bmatrix}. \text{ Το σύνολο δεδομένων, που θα}$$

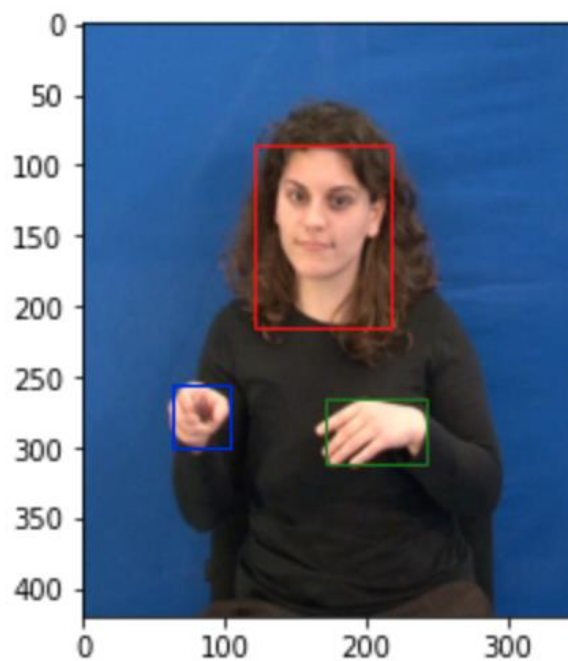
χρησιμοποιήσουμε για την εξαγωγή αυτών των παραμέτρων βρίσκεται στο αρχείο skinSamplesRGB.mat. Τελικά, ως περιοχές δέρματος θα επιλέξουμε τα pixels, για τα οποία ισχύει $P(c) > 0.25$.

Στη συνέχεια, θα εφαρμόσουμε μία μορφολογική επεξεργασία της δυαδικής εικόνας, που προκύπτει από την παραπάνω κατωφλιοποίηση. Αρχικά, θα κάνουμε ένα opening με ένα μικρό δομικό στοιχείο, ώστε να απορριφθούν κάποιοι outliers και, αμέσως μετά, ένα closing με ένα αρκετά μεγαλύτερο δομικό στοιχείο, με σκοπό να αποκτήσουν συνοχή οι περιοχές δέρματος, που

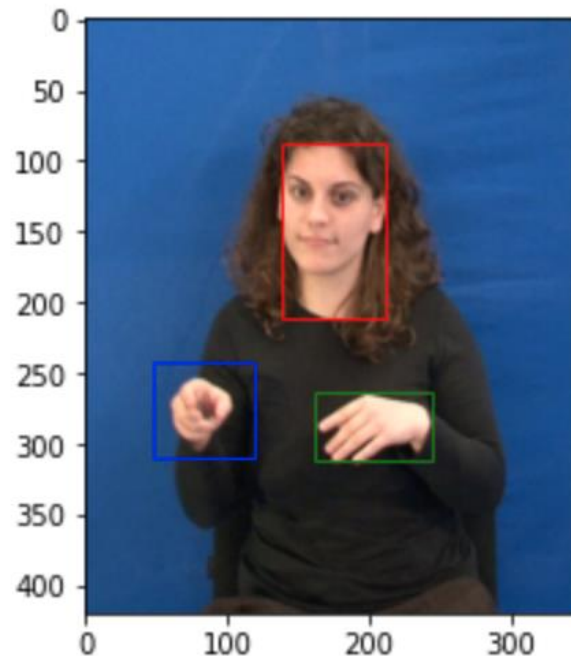
ανιχνεύσαμε. Αριστερά φαίνεται η δυαδική εικόνα πριν τη μορφολογική επεξεργασία και δεξιά φαίνεται η τελική εικόνα:



Επομένως, προκύπτουν τα παρακάτω πλαίσια:



Ωστόσο, για τα επόμενα ερωτήματα θα χρησιμοποιήσουμε κάποια ιδανικά πλαίσια (διευρυμένα σχετικά με τα υπολογισμένα), τα οποία φαίνονται παρακάτω:



1.2 Παρακολούθηση Προσώπου και Χεριών:

Σε αυτό το μέρος θα πραγματοποιηθεί η εκτίμηση της οπτικής ροής του προσώπου και των χεριών της νοηματίστριας με χρήση του αλγορίθμου Lucas-Kanade.

Πρόκειται για έναν επαναληπτικό αλγόριθμο, ο οποίος επιχειρεί να βρει ένα διάνυσμα μετατόπισης d (άρα το πεδίο οπτικής ροής θα είναι $-d$), με το οποίο δύο διαδοχικά frames παρουσιάζουν αντιστοιχία.

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

Η επαναληπτική φύση του αλγορίθμου έγκειται στο γεγονός ότι έχουμε μία εκτίμηση d_i για το d και σε κάθε επανάληψη την βελτιώνουμε κατά u , δηλαδή $d_{i+1} = d_i + u$, μέχρι να συγκλίνει.

Λαμβάνοντας υπόψη ότι ο Lucas-Kanade υπολογίζει τη οπτική ροή σε κάθε σημείο, ελαχιστοποιώντας το παρακάτω τετραγωνικό σφάλμα γύρω από τη γειτονιά του σημείου:

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

Επιπλέον, η ανάλυση κατά Taylor της σχέσης $I_{n-1}(\mathbf{x} + \mathbf{d}) = I_{n-1}(\mathbf{x} + \mathbf{d}_i + \mathbf{u})$, σε συνδυασμό με το παραπάνω σφάλμα μας οδηγούν στον παρακάτω τύπο για το \mathbf{u} για κάθε σημείο της εικόνας:

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

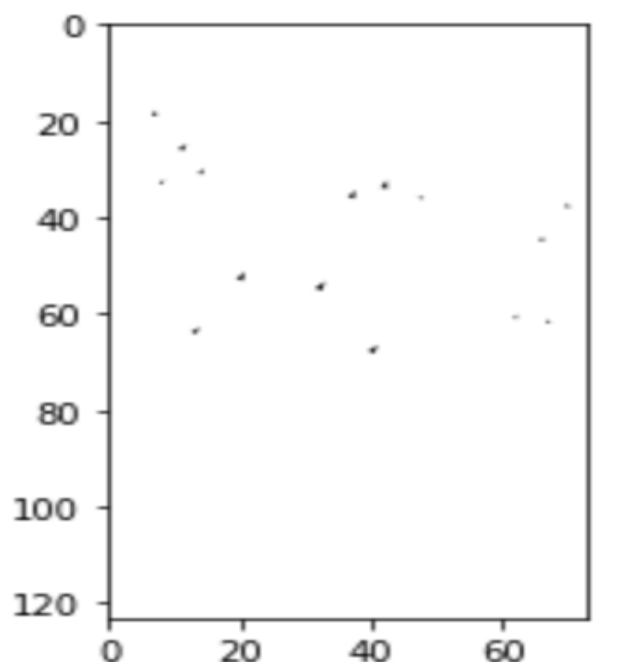
Όπου,

$$A(\mathbf{x}) = \begin{bmatrix} A_1(\mathbf{x}) & A_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial I_{n-1}(\mathbf{x} + \mathbf{d}_i)}{\partial x} & \frac{\partial I_{n-1}(\mathbf{x} + \mathbf{d}_i)}{\partial y} \end{bmatrix}$$

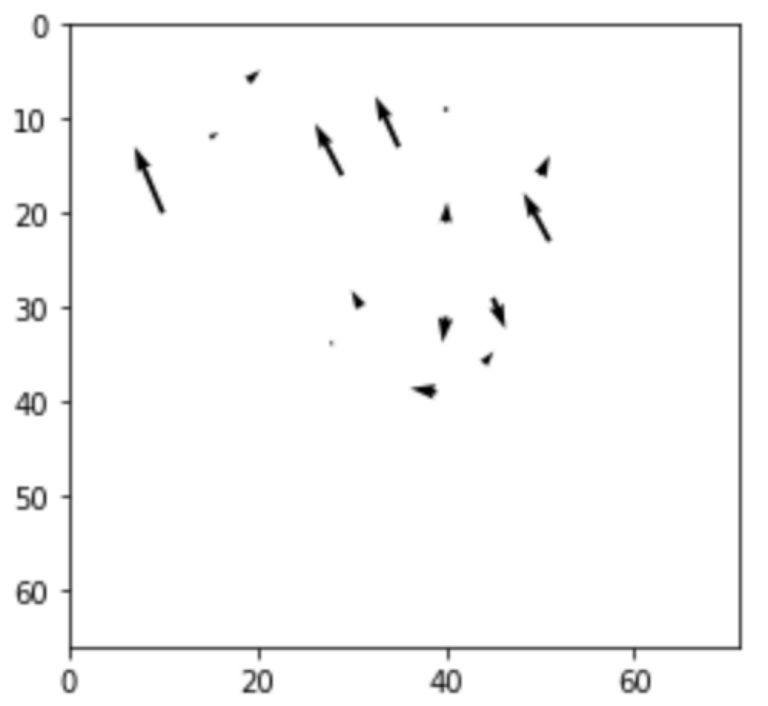
$$E(\mathbf{x}) = I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} + \mathbf{d}_i)$$

Στην υλοποίηση του Lucas-Kanade στην python, η παραπάνω διαδικασία δεν εφαρμόστηκε για κάθε pixel των frames για λόγους υπολογιστικής πολυπλοκότητας, αλλά για συγκεκριμένα σημεία ενδιαφέροντος, τα οποία εξήχθησαν με την μέθοδο των Shi – Tomasi (cv2.goodFeaturesToTrack). Επιπλέον, αφού εξάγουμε τα τελικά διανύσματα κατεύθυνσης για κάθε feature, πραγματοποιείται μία κατωφλιοποίηση με βάση την ενέργειά τους ($\|\mathbf{d}\| = dx^2 + dy^2$), προκειμένου να απορριφθούν ορισμένα διανύσματα που αποτελούν outliers. Η μέση τιμή των εναπομεινανσών διανυσμάτων αποτελεί το συνολικό διάνυσμα μετατόπισης \mathbf{d} . Τα αποτελέσματα που λαμβάνουμε για να τα πρώτα δύο frames είναι τα ακόλουθα:

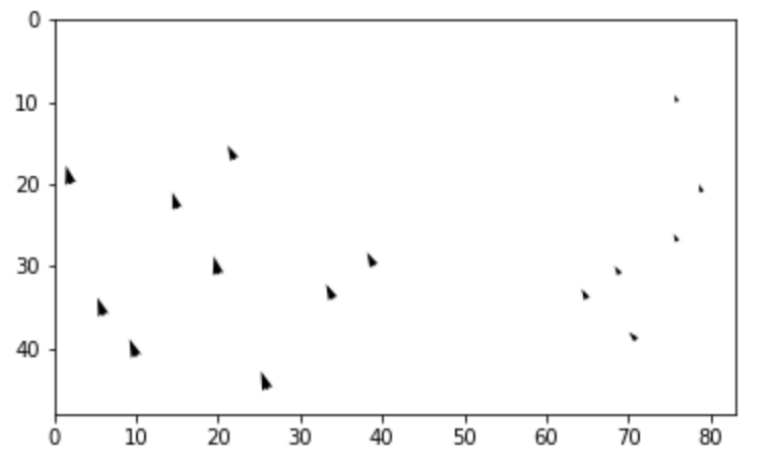
Για το πρόσωπο:



Για το αριστερό χέρι:

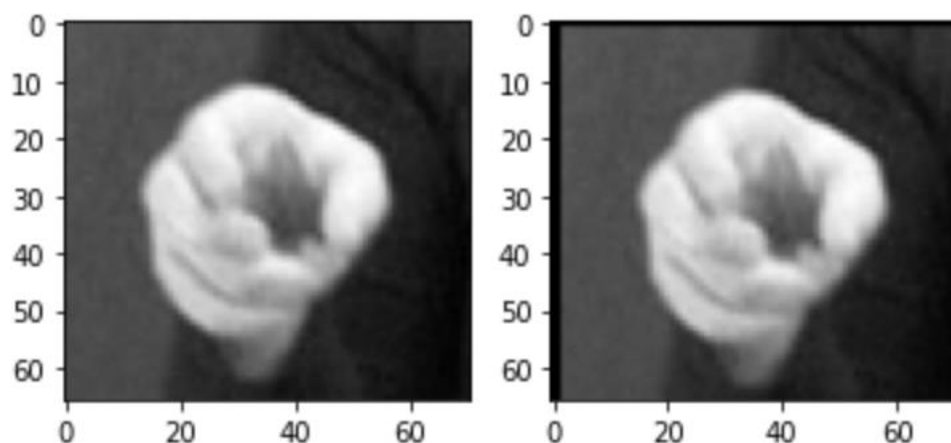


Για το δεξί χέρι:

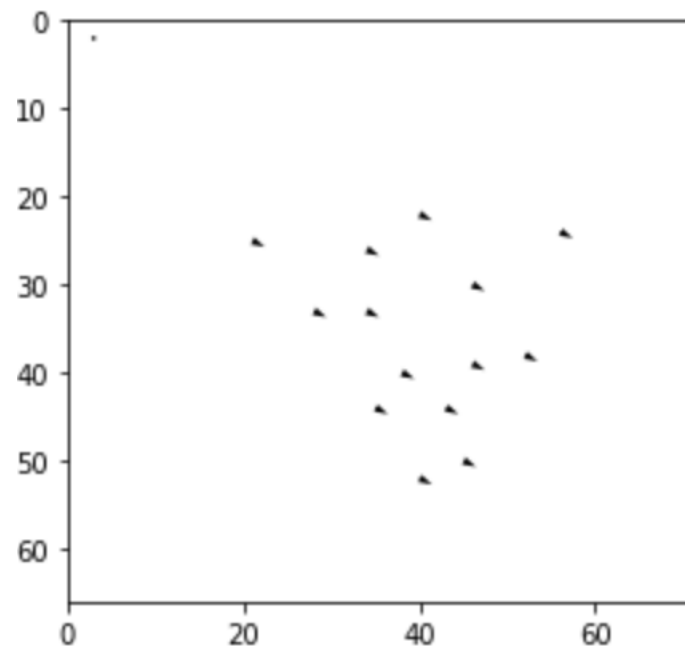


Παρατηρούμε πως, ενώ στο πρόσωπο και στο δεξί έχουμε καλά αποτελέσματα ως προς τη σύγκλιση, κάτι τέτοιο δεν συμβαίνει και στην περίπτωση του αριστερού χεριού. Ο λόγος αυτή της αστοχίας είναι η μεγάλη μεταβολή του αριστερού χεριού από το ένα frame στο επόμενο, με αποτέλεσμα ο αλγόριθμος να το «χάνει». Αντίθετα, τόσο το πρόσωπο, όσο και το δεξί χέρι παρουσιάζουν μικρή μεταβολή (1-2 pixels) και ο αλγόριθμος είναι αποτελεσματικός.

Αξίζει να σημειωθεί πως για την εξέταση της λειτουργικότητας του αλγορίθμου, τον εφαρμόσαμε σε δύο εικόνες, από τις οποίες η δεύτερη ήταν η πρώτη ελαφρώς μετατοπισμένη. Πιο συγκεκριμένα, η μετατόπιση ήταν κατά 1 pixel προς τα κάτω και 2 pixels προς τα δεξιά όπως φαίνεται παρακάτω:



Τα αποτελέσματα που λάβαμε ήταν τα αναμενόμενα, καθώς όλα τα features είχαν τιμές $[-2, -1]$ και το quiver ήταν το ακόλουθο:

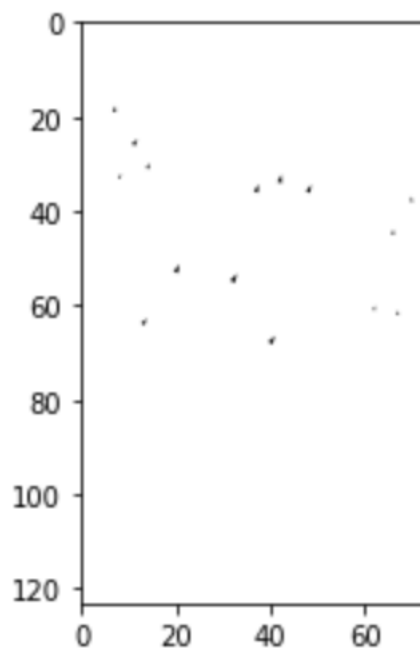


Στη συνέχεια, για να διορθώσουμε την αστοχία, που είχαμε σε μεγάλες μετατοπίσεις, επεκτείναμε την υλοποίηση του Lucas-Kanade σε πολυκλιμακωτή μορφή, ώστε όσο πιο λεπτομερής είναι η εικόνα, τόσο καλύτερη αρχικοποίηση dx_0, dy_0 να έχουμε. Η διαδικασία είναι αρκετά παρόμοια με αυτή, που περιγράψαμε στο προηγούμενο εργαστήριο με τους ανιχνευτές σημείων ενδιαφέροντος.

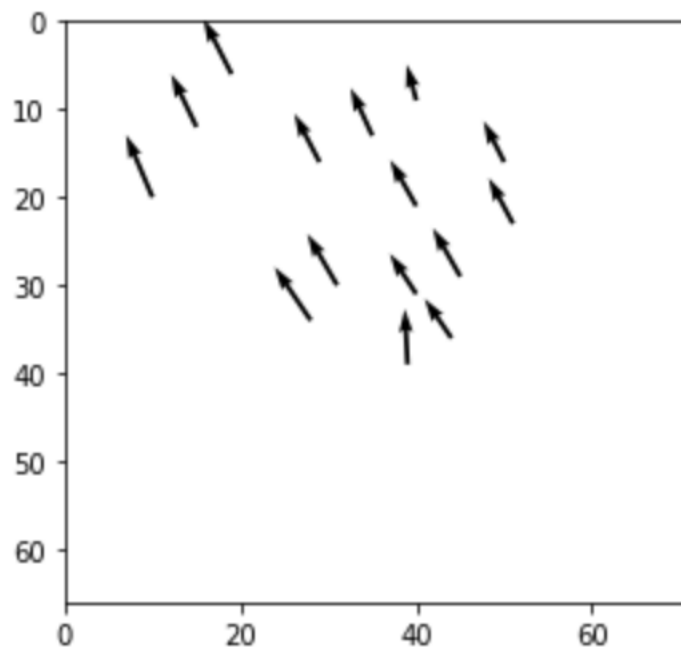
Αρχικά, για να δημιουργήσουμε τις πολλαπλές κλίμακες, εφαρμόζαμε υποδειγματοληψία στην εικόνα της προηγούμενης κλίμακας, αφού, πρώτα, πραγματοποιούσαμε γκαουσιανό φιλτράρισμα, για να αποφύγουμε το aliasing. Έπειτα, από την χαμηλότερη κλίμακα (με αρχικοποίηση $dx_0 = dy_0 = 0$) προς την υψηλότερη (αρχικές διαστάσεις εικόνας), βρίσκαμε σημεία ενδιαφέροντος, τρέχαμε τον Lucas-Kanade και λαμβάναμε το τελικό διάνυσμα μετατόπισης, όπως έχει ήδη περιγραφεί. Στην επόμενη επανάληψη, διπλασιάζαμε το προηγούμενο διάνυσμα μετατόπισης (λόγω της αλλαγής κλίμακας) και το θέταμε ως αρχικοποίηση της επόμενης κλήσης του αλγορίθμου.

Εφαρμόζοντας πολυκλιμακωτό Lucas-Kanade στα ίδια frames με πριν, λάβαμε τα εξής, σαφώς καλύτερα, αποτελέσματα:

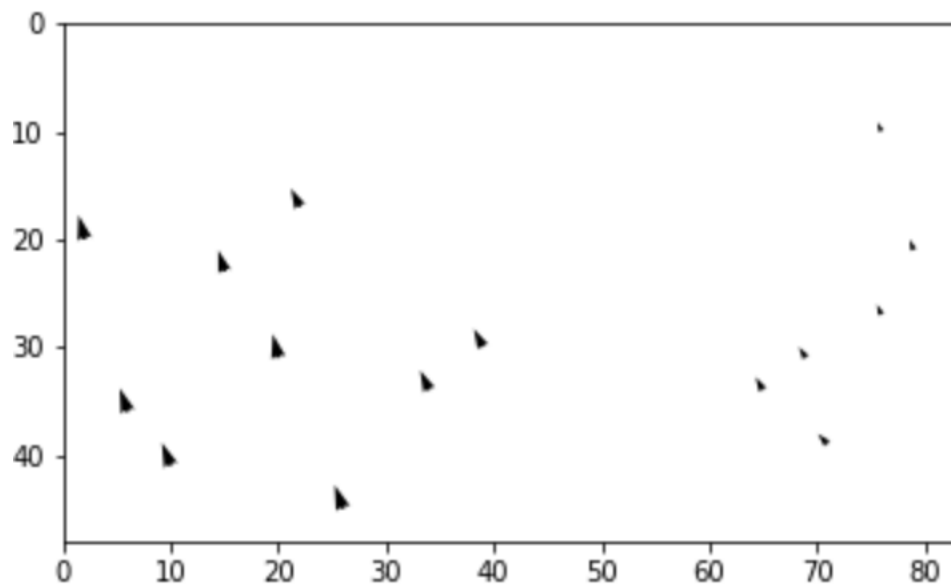
Για το πρόσωπο:



Για το αριστερό χέρι:



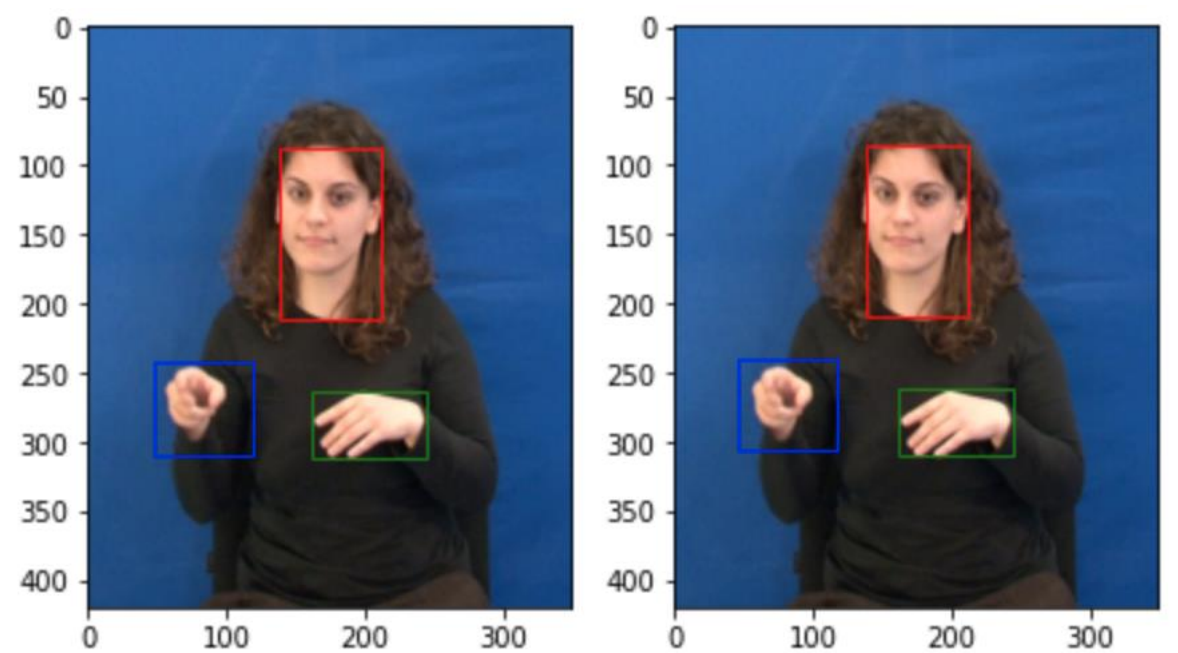
Για το δεξί χέρι:

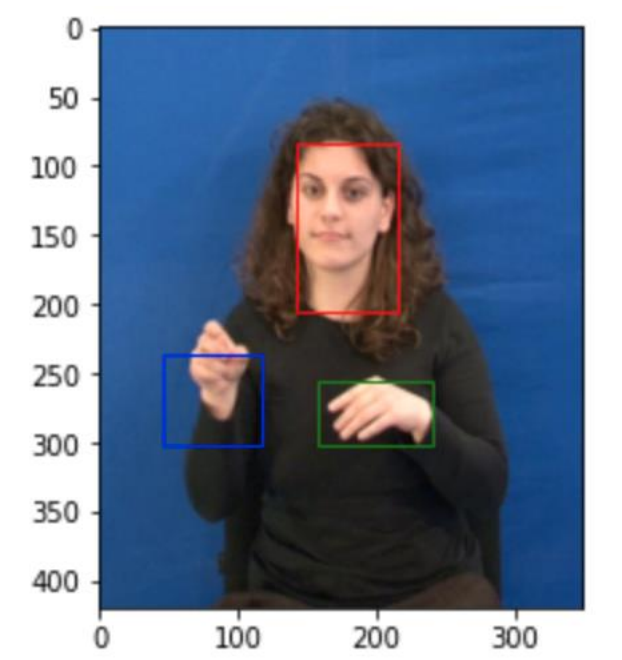
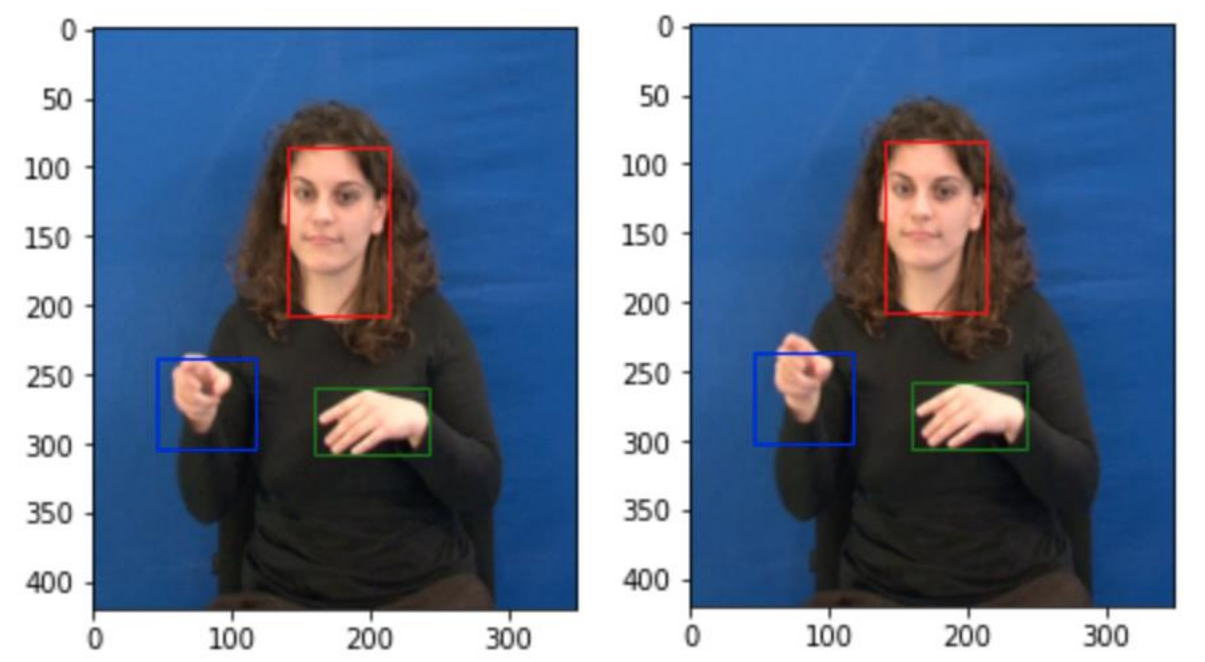


Η κύρια διαφορά, προφανώς, εντοπίζεται στο αριστερό χέρι, η οπτική ροή του οποίου έχει υπολογισθεί με υψηλή ακρίβεια αυτή τη φορά.

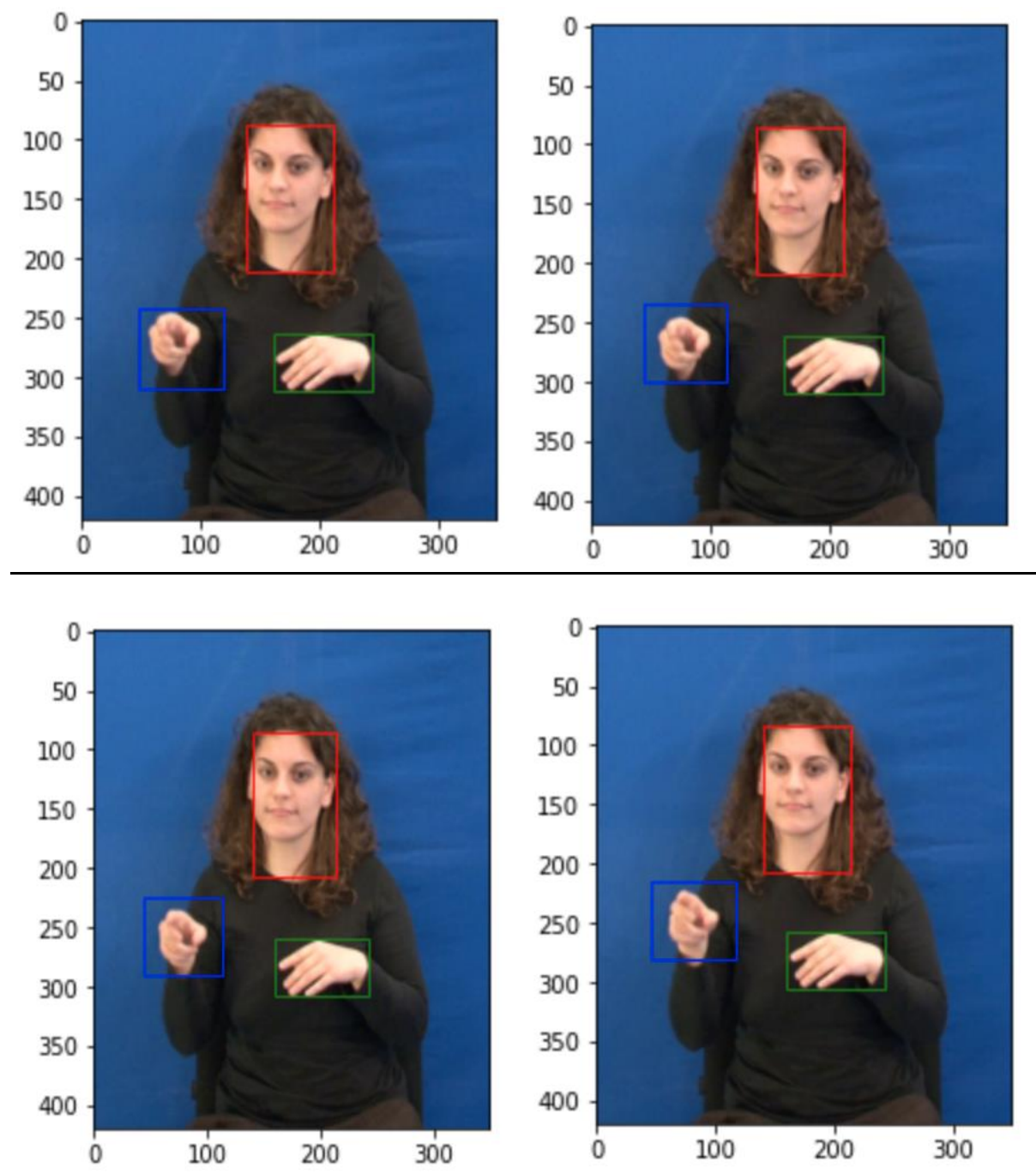
Στη συνέχεια, θα παρουσιαστεί η παρακολούθηση προσώπου και χεριών της νοηματίστριας για τα 5 πρώτα frames, τόσο για τον μονοκλιμακωτό, όσο και τον πολυκλιμακωτό Lucas-Kanade.

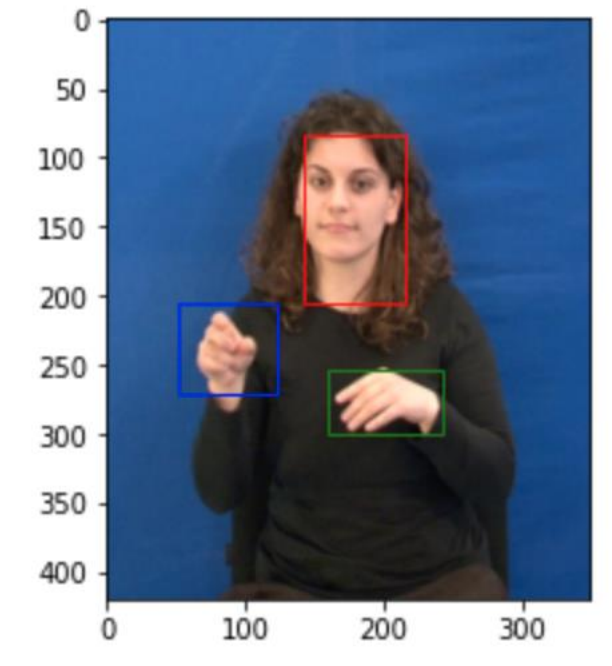
Μονοκλιμακωτός Lucas-Kanade:





Πολυκλιμακωτός Lucas-Kanade:





Όπως φαίνεται στο 3-4 frame η μονοκλιμακωτή έκδοση του αλγορίθμου αρχίζει να «χάνει» το αριστερό χέρι της νοηματίστριας, σε αντίθεση με την πολυκλιμακωτή, η οποία το ακολουθεί κανονικά.

Αξίζει να σημειωθεί, ωστόσο, πως στα επόμενα frames, στα οποία το αριστερό χέρι της νοηματίστριας προσεγγίζει το προσωπό της (μέρος του προσώπου της βρίσκεται στο πλαίσιο του αριστερού χεριού), ακόμα και η πολυκλιμακωτή επέκταση του αλγορίθμου αρχίζει να μην αποδίδει με την αναμενόμενη ακρίβεια.

Σημείωση: Οι παράμετροι που χρησιμοποιήθηκαν για τον Lucas-Kanade στις παραπάνω περιπτώσεις μετά από δοκιμές ήταν οι εξής:

- $\rho = 1$
- $\varepsilon = 0.01$
- $dx_0 = 0$
- $dy_0 = 0$
- $displ_threshold = 1$
- $num_of_scales = 3$

Μέρος 2

2.1 Χωρο-χρονικά Σημεία Ενδιαφέροντος:

Το συγκεκριμένο μέρος αφορά την υλοποίηση των ανιχνευτών χωρο-χρονικών σημείων ενδιαφέροντος Harris και Gabor.

Ο πρώτος συνιστά μία επέκταση του αντίστοιχου ανιχνευτή, που υλοποιήσαμε στην προηγούμενη εργαστηριακή άσκηση στις 3 διαστάσεις.

Πιο συγκεκριμένα, για κάθε frame του βίντεο, θα υπολογίσουμε τον 3x3 πίνακα M ,

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

Και θα εφαρμόσουμε το κριτήριο γωνιότητας:

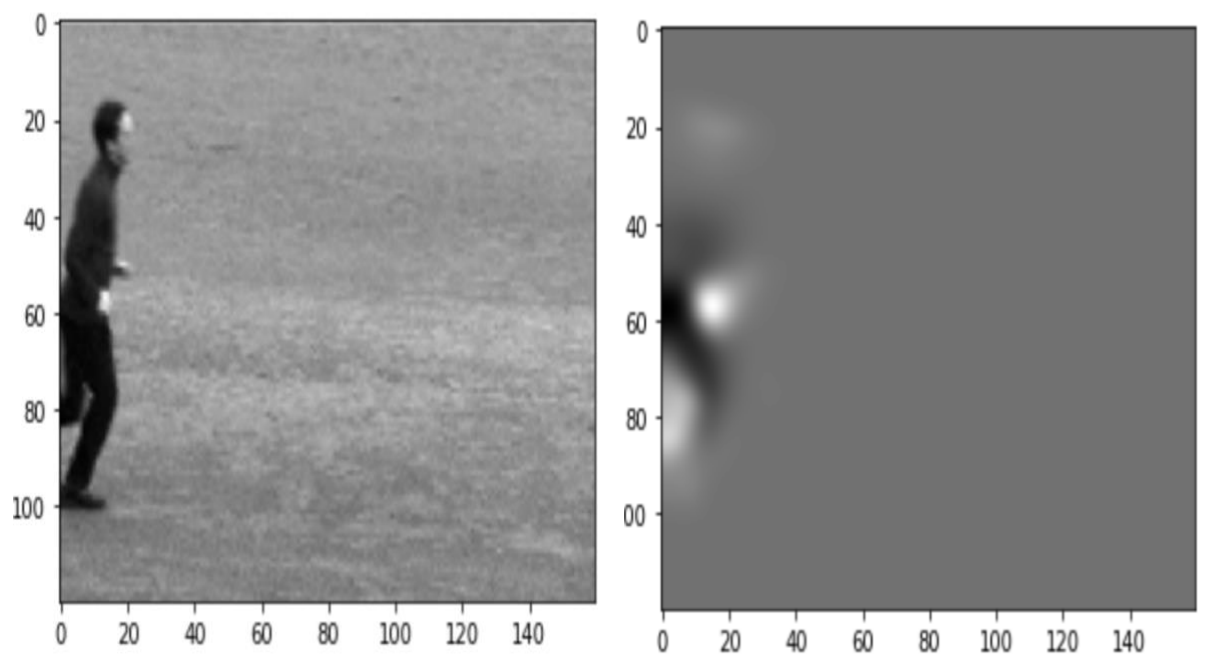
$$H(x, y, t) = \det(M(x, y, t)) - k \cdot \text{trace}^3(M(x, y, t))$$

Τόσο το τρισδιάστατο γκαουσιανό φιλτράρισμα, όσο και η εύρεση των παραγώγων 1^{ης} τάξεως, υλοποιήθηκαν ως επιμέρους μονοδιάστατες συνελίξεις στην αντίστοιχη διάσταση (`ndi.convolve1d`), με σκοπό να μειωθεί η υπολογιστική πολυπλοκότητα του ανιχνευτή.

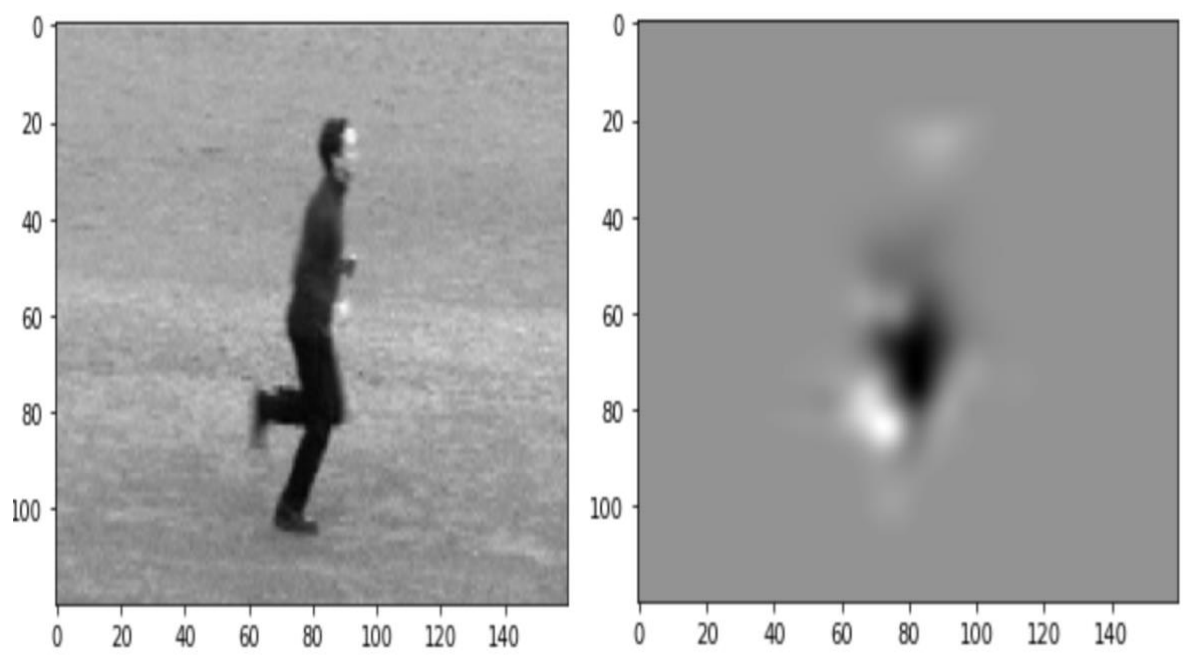
Στη συνέχεια, επαναλάβουμε το κριτήριο επιλογής τοπικών μεγίστων και κατωφλιοποίησης σε κάθε frame, προκειμένου να εξάγουμε τα τελικά σημεία ενδιαφέροντος. Από αυτά θα κρατήσουμε τα 500 καλύτερα.

Αξίζει να οπτικοποιήσουμε το κριτήριο H για ορισμένα frames μιας τυχαίας εικόνας, καθώς παρουσιάζει ιδιαίτερο ποιοτικό ενδιαφέρον. Στις παρακάτω εικόνες φαίνεται πως το ενδιαφέρον σε έναν frame ενός ανθρώπου, που τρέχει εστιάζεται, όπως είναι φυσιολογικό, στα χέρια και στα πόδια, τα οποία έχουν την εντονότερη κίνηση.

Για το frame 100:



Για το frame 110:



Ο ανιχνευτής Gabor προκύπτει από την τετραγωνική ενέργεια της εξομαλυμένης εικόνας για το ζεύγος των ορθογώνιων Gabor συνιστωσών,

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

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

Επομένως, η τελική μαθηματική έκφραση, που χαρακτηρίζει τον ανιχνευτή είναι η ακόλουθη:

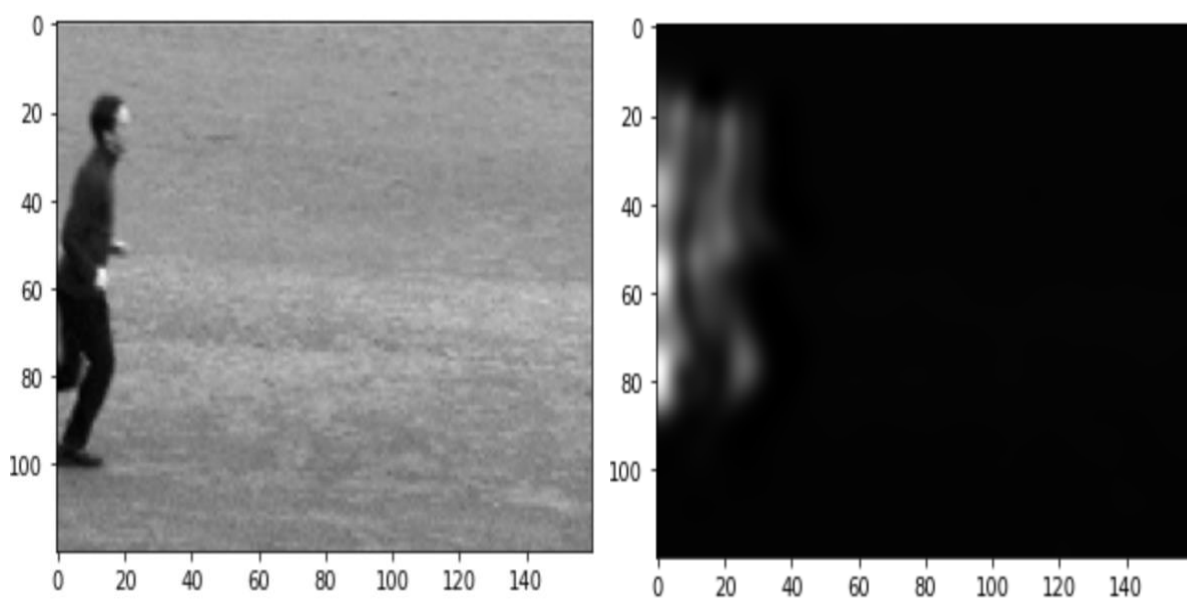
$$H(x, y, t) = (I(x, y, t) * g * h_{ev})^2 + (I(x, y, t) * g * h_{od})^2$$

Όπου g δισδιάστατος γκαουσιανός πυρήνας τυπικής απόκλισης σ .

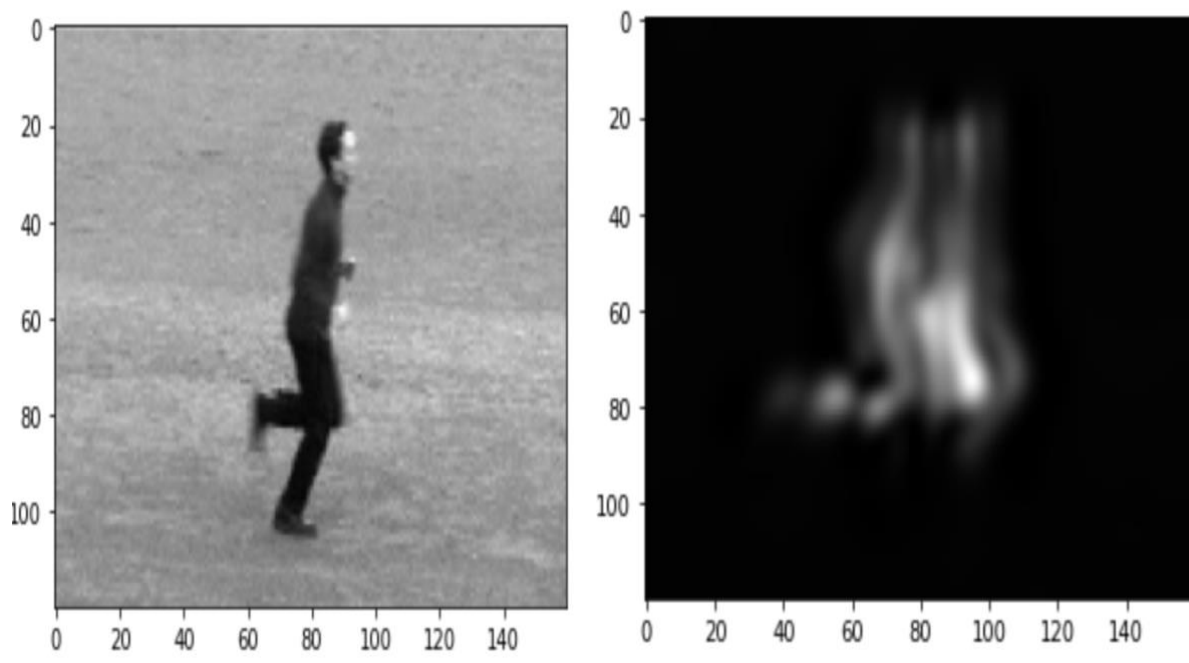
Αξίζει να σημειωθεί πως πρώτου χρησιμοποιήσουμε τα δύο φίλτρα Gabor για την εξαγωγή του κριτηρίου γωνιότητας, τα κανονικοποιούμε με την L1 νόρμα τους.

Η επιλογή τοπικών μεγίστων γίνεται με τον ίδιο τρόπο, που περιγράψαμε για τον προηγούμενο ανιχνευτή. Οι αντίστοιχες οπτικοποιήσεις του κριτηρίου H φαίνονται παρακάτω:

Για το frame 100:



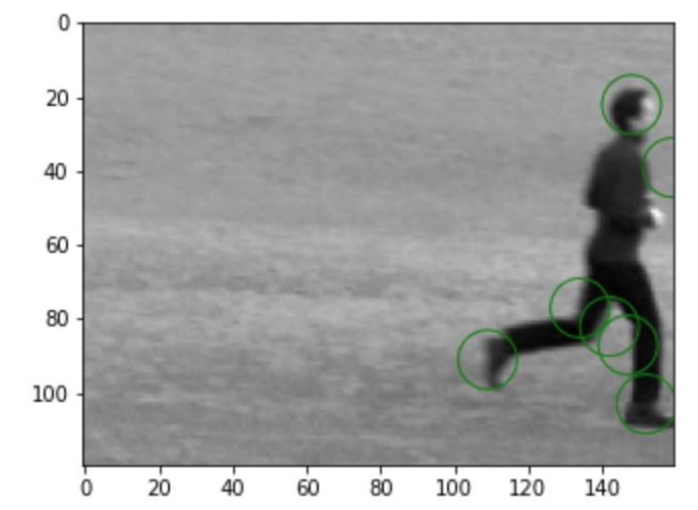
Για το frame 110:



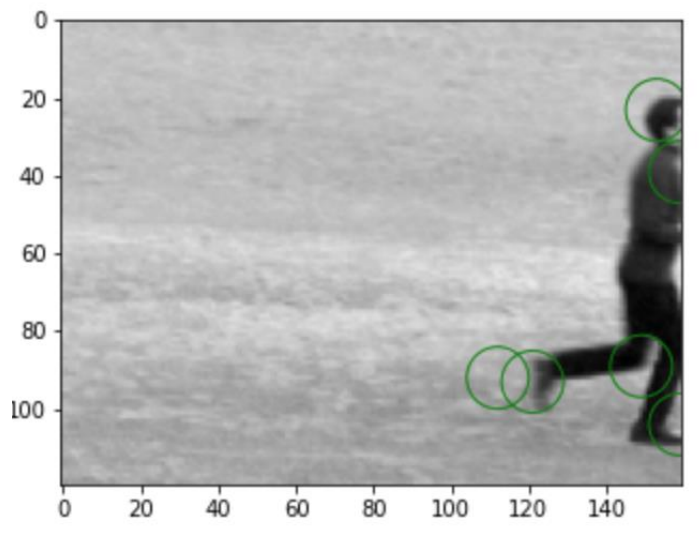
Συγκρίνοντας τα κριτήρια γωνιότητας των δύο ανιχνευτών επιβεβαιώνεται το γεγονός πως ο Gabor ανιχνευτής είναι αρκετά πιο ευαίσθητος στην επιλογή σημείων ενδιαφέροντος.

Παρακάτω φαίνονται δύο διαδοχικά frames ενός video. Στο πρώτο απεικονίζονται τα σημεία ενδιαφέροντος με χρήση του ανιχνευτή Harris και στο με χρήση του ανιχνευτή Gabor

Harris Detector for frame 118:



Gabor Detector for frame 119:



Στον φάκελο part2_videos περιλαμβάνονται 6 videos με κάθε συνδυασμό των δύο ανιχνευτών με ένα τυχαίο video κάθε μιας από τις τρεις δράσεις (running, walking, boxing).

Σημείωση: Ύστερα από δοκιμές οι παράμετροι, που χρησιμοποιήθηκαν για την δημιουργία των videos, αλλά και των τοπικών περιγραφητών στη συνέχεια είναι οι εξής:

- $\sigma = 4$
- $\tau = 1.5$
- $k = 0.005$ (Harris)
- $s = 1$ (Harris)

2.2 Χωρο-χρονικοί Ιστογραφικοί Περιγραφητές:

Το επόμενο βήμα μετά την ανίχνευση σημείων ενδιαφέροντος είναι η εξαγωγή χαρακτηριστικών για τα videos. Η διαδικασία, αυτή, όπως και στην προηγούμενη εργαστηριακή άσκηση, θα πραγματοποιηθεί με χρήση τοπικών περιγραφητών.

Η πρώτη περίπτωση περιγραφητών που θα χρησιμοποιηθεί είναι οι HOG (Histogram of Oriented Gradients), που αφορούν το διάνυσμα κλίσης σε κάθε pixel, που βρίσκεται στις γειτονιές (γειτονιά 4σ) κάθε ενός από τα σημεία ενδιαφέροντος, που έχουμε ήδη ανιχνεύσει. Η δεύτερη περίπτωση είναι οι HOF (Histogram of Oriented Flow), οι οποίοι δημιουργούνται με τον ίδιο τρόπο με τους HOG περιγραφητές, αλλά αφορούν την οπτική ροή σε κάθε pixel.

Για την παραπάνω διαδικασία εξαγωγής περιγραφητών θα χρησιμοποιηθεί συνάρτηση `orientation_histogram`, που περιλαμβάνεται στη αρχείο `cv22_lab2_2_utlis.py`.

Τέλος, θα δημιουργήσουμε και μία τρίτη περίπτωση περιγραφητών, η οποία θα αποτελείται από απλό concatenation των περιγραφητών HOG και HOF για κάθε σημείο ενδιαφέροντος. Για αυτό τον λόγο ονομάζονται HOG/HOF περιγραφητές.

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

Στη συνέχεια, θα διαχωρίσουμε το σύνολο των 48 videos σε train και test με 36 και 12 videos, αντίστοιχα. Αυτή η διαμέριση θα γίνει σύμφωνα με το αρχείο `training_videos.txt` και τα τελικά train και test σύνολα, θα περιέχουν τους περιγραφητές για τα αντίστοιχα videos.

Ακολουθεί ο υπολογισμός του bag of words των περιγραφητών, ο οποίος θα πραγματοποιηθεί με την έτοιμη συνάρτηση `bag_of_words`, που δέχεται ως ορίσματα τους train και test περιγραφητές, καθώς και τον αριθμό κέντρων του k-means clustering, που θα ακολουθήσει.

Τέλος, θα γίνει η τελική ταξινόμηση των test videos μέσω ενός SVM (Support Vector Machine), με τη συνάρτηση `svm_train_test`. Για τους

6 συνδυασμούς των περιγραφητών και ανιχνευτών, προκύπτει το ακόλουθο πίνακάκι με τα accuracies:

	HOG	HOF	HOGHOF
Harris	0.9167	0.5833	0.6667
Gabor	0.6667	0.5	0.5

Παρατηρείται πως ο καλύτερος συνδυασμός επιτυγχάνεται με ανιχνευτή Harris και περιγραφητές HOG.

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. Παρόλα αυτά, λάθη πραγματοποιούνται και με το boxing, αν και αυτή η δράση αφορά, κυρίως, τα χέρια, σε αντίθεση με τις άλλες δύο. Χαρακτηριστικό παράδειγμα είναι ότι το μοναδικό misclassification του συνδυασμού Harris και HOG ήταν ένα βίντεο boxing, που ταξινομήθηκε ως running.

Μέρος 3

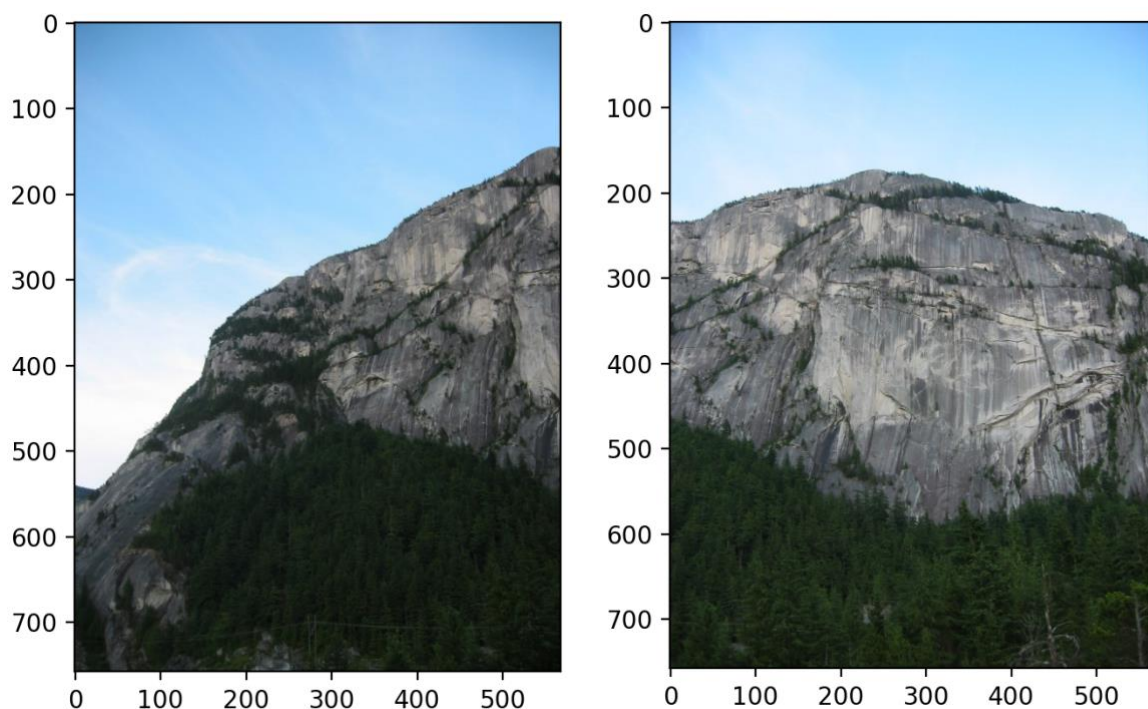
Συνένωση εικόνων για δημιουργία πανοράματος:

Στο συγκεκριμένο μέρος θα επιχειρήσουμε να συνενώσουμε εικόνες, που έχουν ληφθεί από το ίδιο μέρος του πραγματικού κόσμου με απλή περιστροφή της κάμερας, με σκοπό τη δημιουργία ενός πανοράματος.

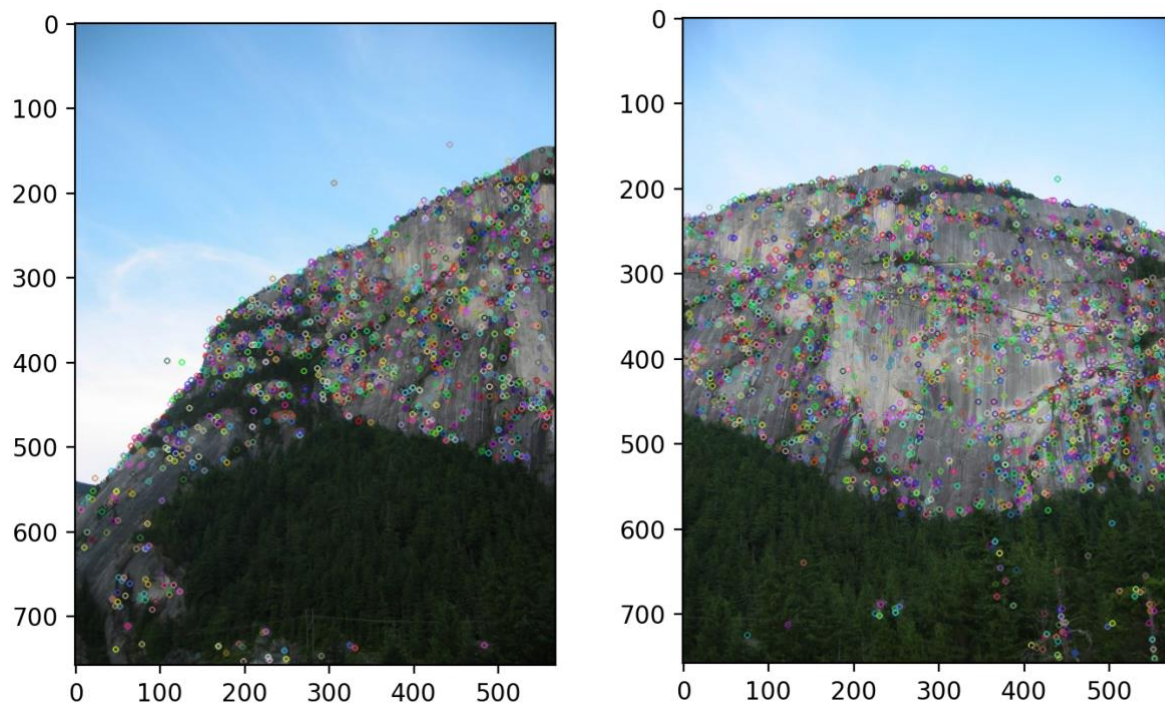
Ο αλγόριθμος που θα υλοποιήσουμε χωρίζεται στα εξής βήματα:

1. Ανίχνευση σημείων ενδιαφέροντος και εξαγωγή περιγραφητών για τις δύο εικόνες προς συνένωση.
2. Ταίριασμα των εντοπισμένων σημείων.
3. Διατήρηση των καλύτερων ταιριασμάτων μέσω εφαρμογής του κριτηρίου Lowe.
4. Υπολογισμός ομογραφίας με RANSAC.
5. Υπολογισμός μετασχηματισμένης εικόνας με τη χρήση της ομογραφίας και της τεχνικής του Inverse Warping.
6. Συνένωση των εικόνων.

Οι δύο πρώτες εικόνες για συνένωση:

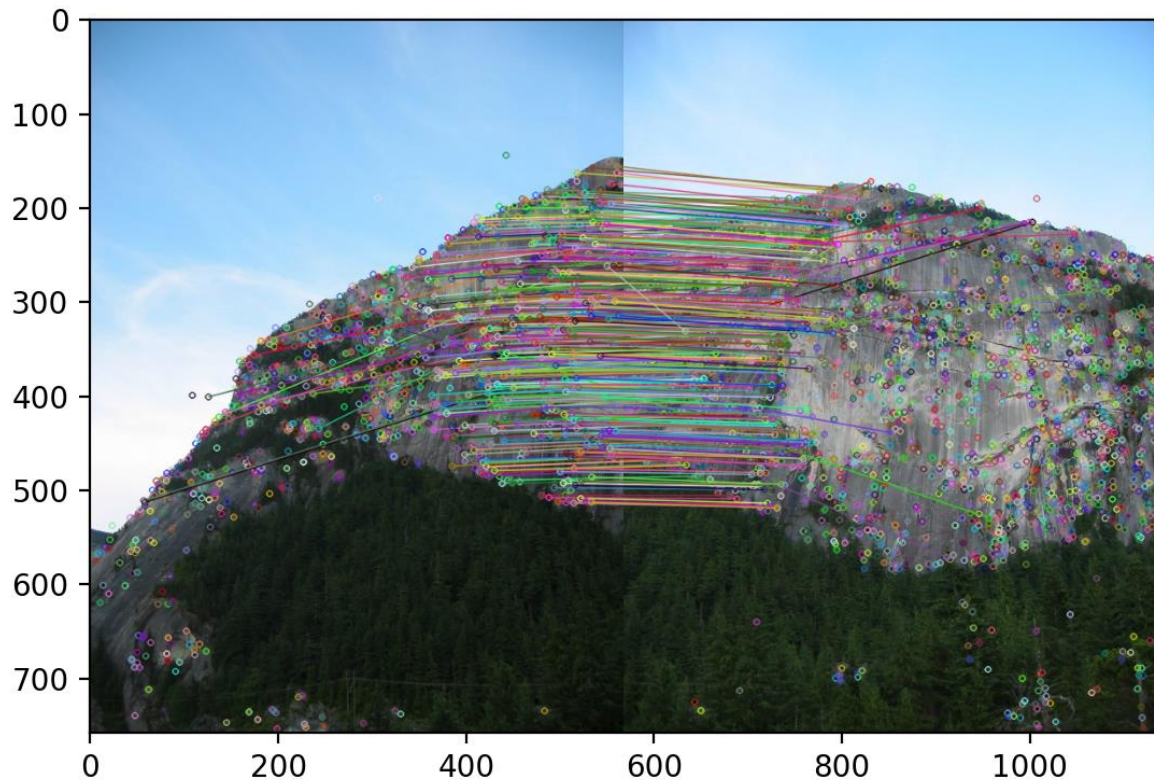


Επομένως, αρχικά, θα υπολογίσουμε τα σημεία ενδιαφέροντος και θα εξάγουμε τους SIFT περιγραφητές, μέσω της συνάρτησης `detectAndCompute` της κλάσης `SIFT_create`. Τα εντοπισμένα σημεία φαίνονται παρακάτω:



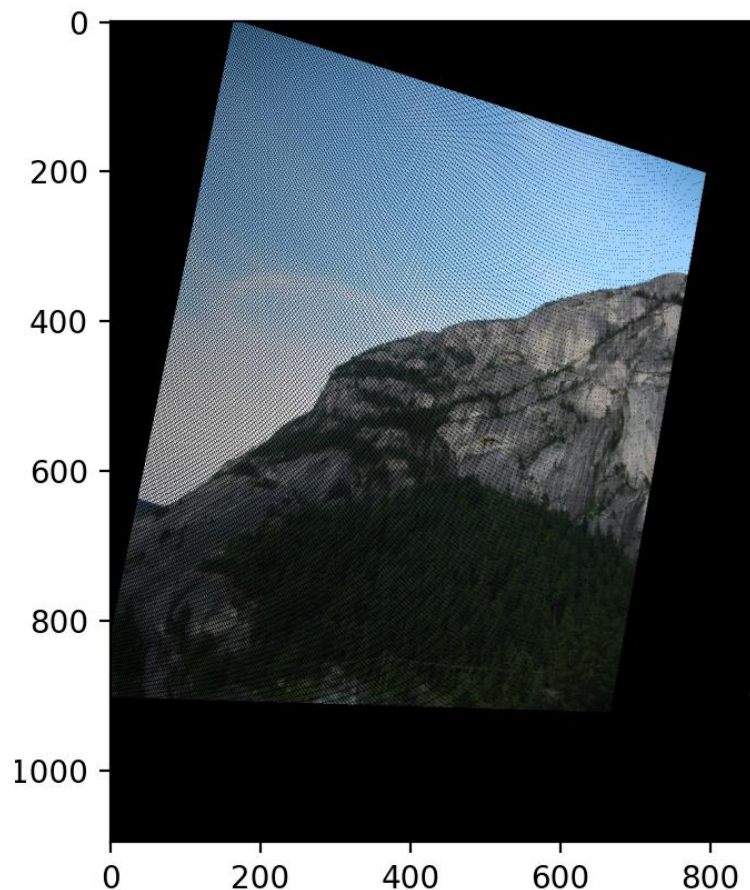
Στη συνέχεια, για τη μείωση της υπολογιστικής πολυπλοκότητας της διαδικασίας, θα λάβουμε τους 2, προσεγγιστικά, κοντινότερους γείτονες, για κάθε σημείο ενδιαφέροντος χρησιμοποιώντας τη βιβλιοθήκη FLANN (Fast Library for Approximate Nearest Neighbors). Έχοντας τις αποστάσεις των 2, προσεγγιστικά, κοντινότερων γειτόνων θα, προκειμένου να απορρίψουμε ένα μεγάλο πλήθος λάθος αντιστοιχίσεων, θα ακολουθήσουμε το κριτήριο Lowe. Σύμφωνα με αυτό, διατηρούμε μόνο τις αντιστοιχίσεις, στις οποίες ο λόγος των αποστάσεων του κοντινότερου γείτονα με τον δεύτερο κοντινότερο γείτονα είναι μικρότερη του 0.8. Με αυτό τον τρόπο, έχει αποδειχθεί πειραματικά ότι απορρίπτουμε το 90% των λανθασμένων αντιστοιχίσεων και, συγχρόνως, μόνο το 5% των σωστών.

Μετά από αυτό το «φιλτράρισμα», προκύπτουν οι εξής τελικές αντιστοιχίσεις:



Διατηρώντας το βελτιωμένο σύνολο αντιστοιχίσεων, θα βρούμε μέσω της `cv2.findHomography`, με όρισμα `method = RANSAC`, την ομογραφία H για την οποία ισχύει η σχέση $x_2 = Hx_1$.

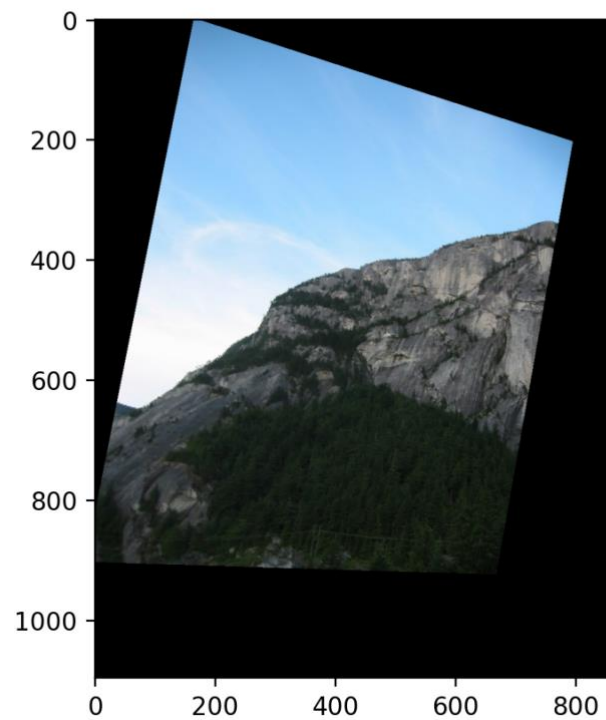
Με αυτό τον μετασχηματισμό (και την διαίρεση με w για τις ομογενείς συντεταγμένες) θα έχουμε την τελική warped εικόνα 1, η οποία είναι η ακόλουθη:



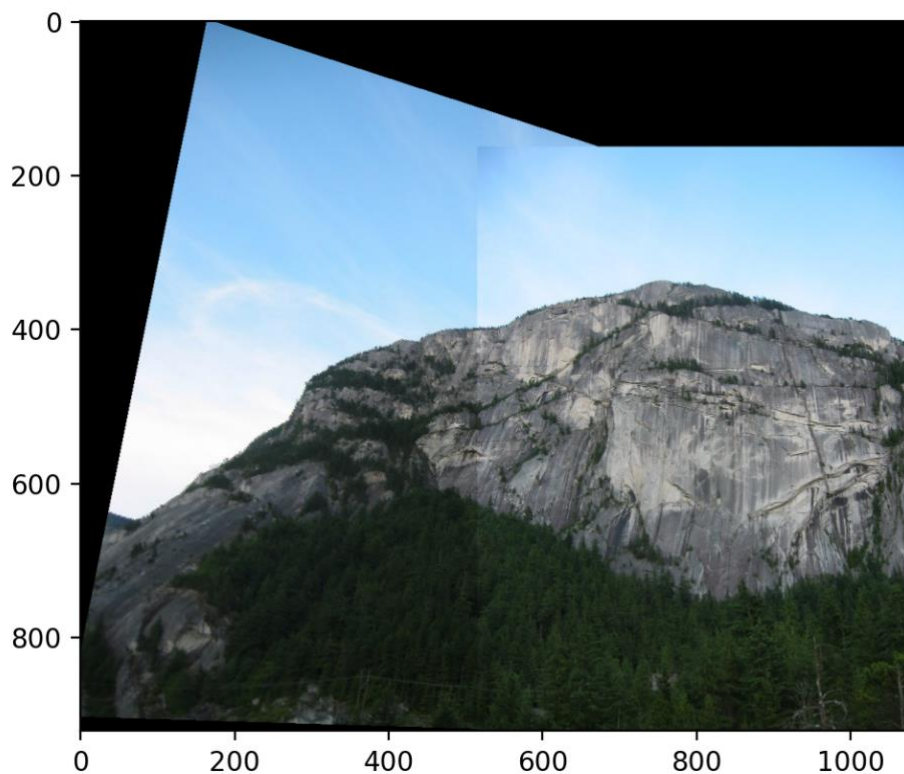
Παρόλο που έχουμε τον αναμενόμενο μετασχηματισμό της εικόνας, παρατηρούνται κάποια «κενά» μέσα στην εικόνα, καθώς έχουμε pixels με μηδενική τιμή. Αυτό συμβαίνει διότι από την ομογραφία δεν λαμβάνουμε ακριβώς 1-1 αντιστοίχιση των pixels, επιστρέφοντας float αριθμοί, τους οποίους χειριζόμαστε με round, floor ή ceil.

Για αυτό τον λόγο, πολλές φορές μένουν κενά μερικά pixels, ενώ σε μερικά άλλα γίνεται και overwrite, το οποίο, όμως, δεν καθιστά πρόβλημα, λόγω των πολύ κοντινών (και εξαρτώμενων) τιμών των γειτονικών pixels. Προκειμένου να αποφύγουμε τα κενά pixels, θα εφαρμόσουμε inverse warping, βρίσκοντας τον αντίστροφο πίνακα της ομογραφίας H^{-1} και αντιστοιχίζοντας τα μηδενικά με τα pixels της αρχικής εικόνας.

Με την εφαρμογή αυτής της τεχνικής, λαμβάνουμε την τελική εικόνα:



Το τελευταίο βήμα το αλγορίθμου είναι απλά να συνενώσουμε την warped εικόνα με τη δεύτερη ως προς τις συντεταγμένες της δεύτερης. Το τελικό αποτέλεσμα, που προκύπτει είναι το ακόλουθο:



Επαναλαμβάνουμε τη διαδικασία για τις εικόνες 1,2,3 και για τις 4,5,6 και συνενώνουμε τις δύο επιμέρους συνενώσεις, για να παραχθεί το ζητούμενο πανόραμα:

