



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΜΑΘΗΣΗΣ

Ομάδα M24:

1^ο μέλος: Ευαγγελία Ρεντίφη, Α.Μ.: 03116164

2^ο μέλος: Ειρήνη Κληρονόμου, Α.Μ.: 03116094

Τεχνολογία Και Ανάλυση Εικόνων και Βίντεο

Άσκηση 1: Επεξεργασία εικόνας, φίλτρα, ακμές και εκτίμηση κίνησης

Ερώτημα 2

Στο ερώτημα αυτό θα χρησιμοποιήσουμε αλγορίθμους εξαγωγής χαρακτηριστικών, καθώς και τεχνικές εκτίμησης κίνησης, για να εντοπίσουμε τα κινούμενα στοιχεία σε surveillance videos από το VIRAT dataset. Ο κώδικας της άσκησης βρίσκεται στο αρχείο `ergasia1.py`. Για την υλοποίηση του χρησιμοποιήθηκε το `pycharm 2019.2.4`, όπως προτάθηκε από τους διδάσκοντες. Για να τρέξετε τον κώδικα πιθανώς να χρειαστεί να αλλάξετε τα `paths`. Εμείς για ευκολία μετονομάσαμε το βίντεο σε `dromos.mp4` και το είχαμε αποθηκεύσει στον τοπικό δίσκο (C:) του υπολογιστή μας.

➤ Στο **ερώτημα 3** εφαρμόσαμε Harris και Shi-Tomasi corners detectors στο πρώτο frame του δοθέντος βίντεο. Για το σκοπό αυτό χρησιμοποιήθηκε η συνάρτηση `cv2.goodFeaturesToTrack` με τα εξής ορίσματα:

image: είσοδος 8-bit ή floating-point 32-bit, μονοκάναλη εικόνα

mask : περιορίζει την περιοχή στην οποία θα ανιχνευθούν τα Harris ή Shi Tomasi σημεία ενδιαφέροντος. Εμείς θέσαμε `mask=None` καθώς θέλουμε να εντοπίσουμε τις γωνιές σε όλη την έκταση του frame.

maxCorners: Μέγιστος αριθμός των γωνιών που θα επιστραφούν

qualityLevel: Παράμετρος που χαρακτηρίζει την ελάχιστη αποδεκτή ποιότητα των γωνιών της εικόνας. Η τιμή της παραμέτρου πολλαπλασιάζεται με το καλύτερο μέτρο ποιότητας γωνίας, το οποίο είναι η ελάχιστη τιμή `eigen` (βλ. `CornerMinEigenVal ()`) ή την απόκριση συνάρτησης Harris (βλ. `CornerHarris ()`). Οι γωνίες με το μέτρο ποιότητας λιγότερο από το προϊόν απορρίπτονται. Για παράδειγμα, εάν η καλύτερη γωνία έχει το μέτρο ποιότητας = 1500 και το επίπεδο ποιότητας = 0,01, τότε απορρίπτονται όλες οι γωνίες με το μέτρο ποιότητας μικρότερο από 15

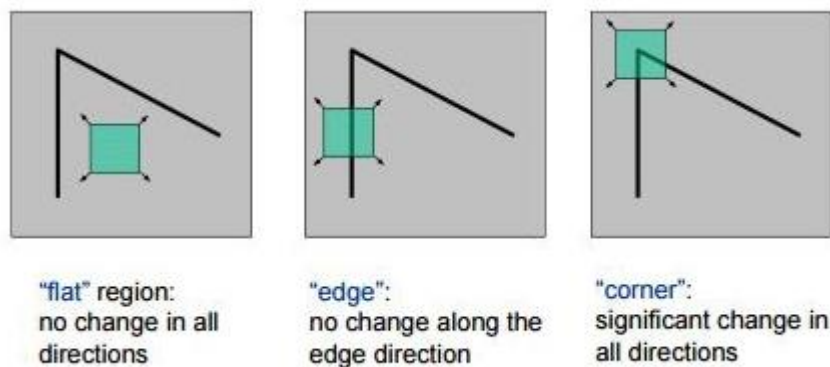
minDistance: Η μικρότερη δυνατή ευκλείδεια απόσταση μεταξύ των επιστρεφόμενων γωνιών.

blockSize: Μέγεθος ενός μέσου μπλοκ για μιας παράγωγης μήτρα συνδιακύμανσης σε κάθε γειτονιά εικονοστοιχείων.

useHarrisDetector: Παράμετρος που δείχνει αν χρησιμοποιείται ο Harris corner detector.

(Τα παραπάνω βρέθηκαν από το σχετικό documentation της συνάρτησης)

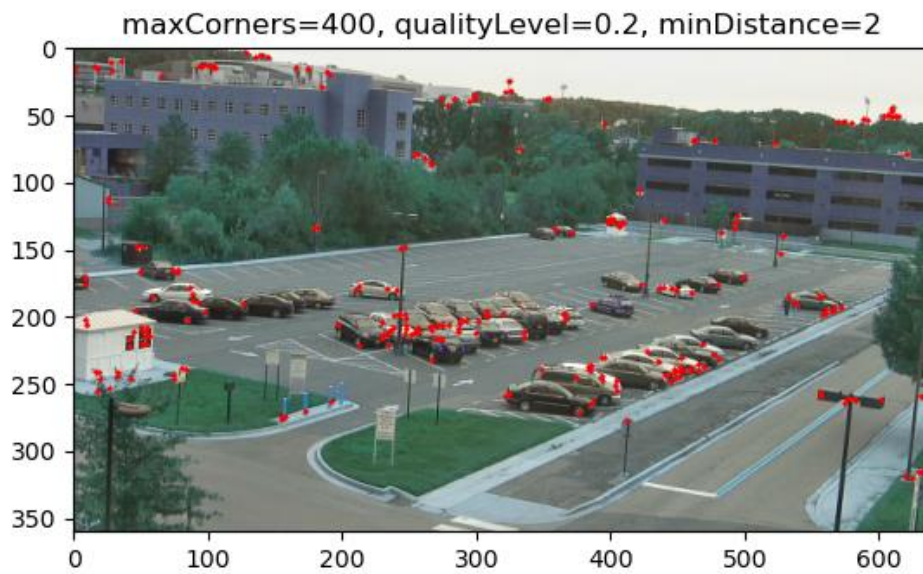
Επίσης να σημειώσουμε ότι **με τον όρο corner(γωνία)** προσδιορίζουμε ένα σημείο του οποίου η τοπική γειτονιά βρίσκεται σε δύο κυρίαρχες και διαφορετικές ακμές. Οι γωνίες είναι τα σημαντικά χαρακτηριστικά της εικόνας και γενικά ορίζονται ως σημεία ενδιαφέροντος που είναι αναλλοίωτα στη μετατόπιση, την περιστροφή και το φωτισμό



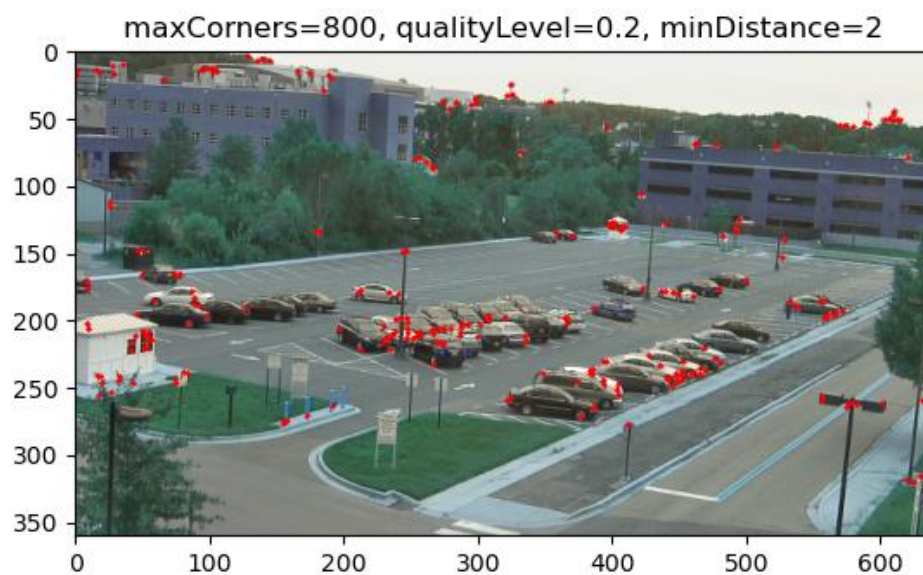
Εάν πάρουμε την επίπεδη περιοχή της εικόνας (δείτε την προηγούμενη εικόνα) τότε δεν παρατηρείται αλλαγή κλίσης προς οποιαδήποτε κατεύθυνση. Ομοίως, στην περιοχή των ακμών, δεν παρατηρείται αλλαγή κλίσης κατά την κατεύθυνση της ακμής. Στη γωνιακή περιοχή παρατηρούμε μια σημαντική αλλαγή κλίσης προς όλες τις κατευθύνσεις.

Εμείς πειραματιστήκαμε με τις παραμέτρους maxCorners, qualityLevel και minDistance της συνάρτησης `cv2.goodFeaturesToTrack` τόσο για shi tomasi όσο και για harris corners ώστε να οδηγηθούμε στο βέλτιστο αποτέλεσμα(να αποτυπώσουμε δλδ όσο το δυνατόν καλύτερα τις γωνίες του πρώτου frame). Παρακάτω φαίνονται τα σχετικά screenshots.

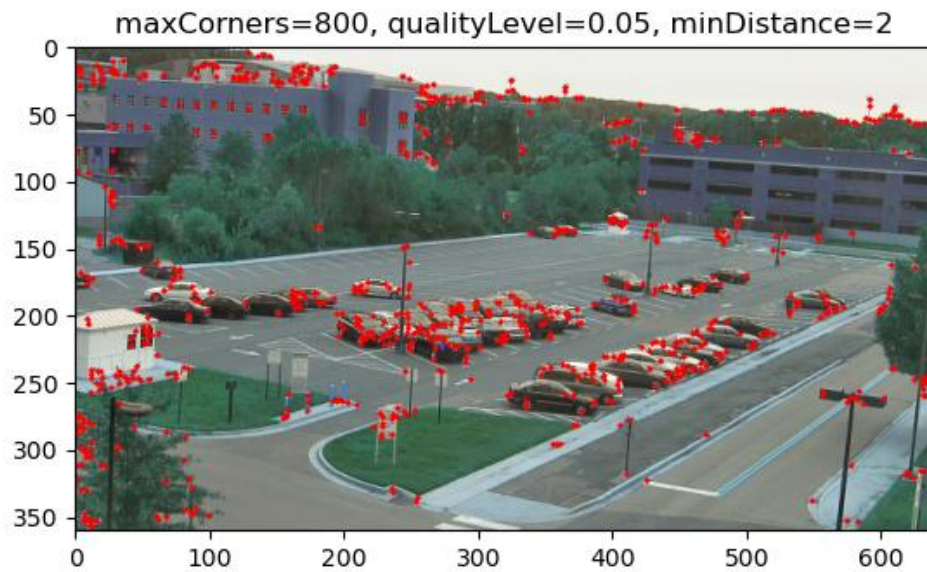
Shi Tomasi



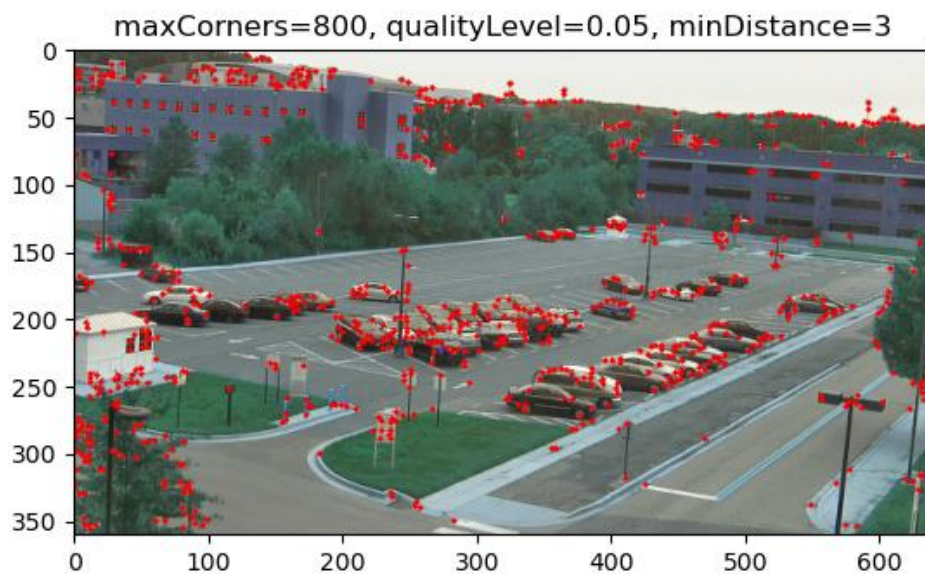
Παρατηρούμε ότι για τις συγκεκριμένες τιμές των παραμέτρων δεν έχουμε καλή αποτύπωση των γωνιών του πρώτου frame.



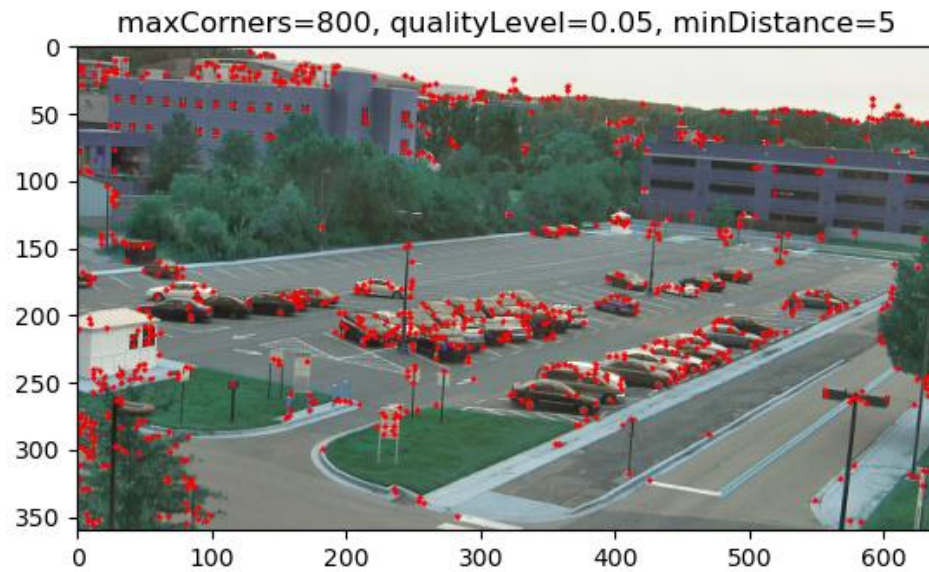
Έπειτα αυξάνουμε τον αριθμό των επιστρεφόμενων γωνιών (θέτουμε maxCorners=800 αντί για 400, κρατώντας σταθερές τις τιμές των υπόλοιπων παραμέτρων) αλλά δεν παρατηρούμε σημαντικές μεταβολές.



Στη συνέχεια μεταβάλλουμε την τιμή του qualityLevel και την θέτουμε ίση με 0.05 και παρατηρούμε ότι εμφανίζονται αρκετά σημεία ενδιαφέροντος στην εικόνα μας, ωστόσο αρκετά από αυτά είναι αρκετά κοντά και δεν διακρίνονται το ένα από το άλλο.



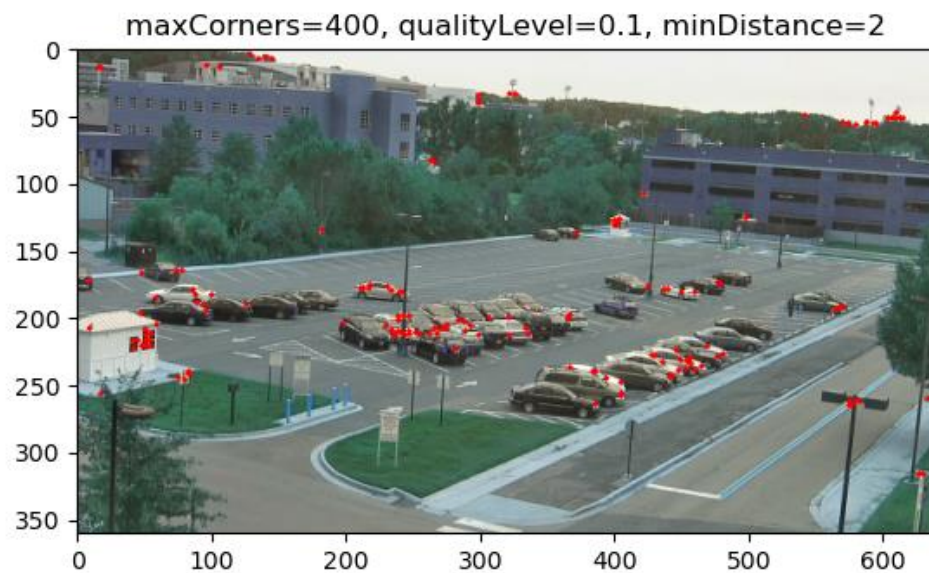
Οπότε αύξανουμε την μικρότερη δυνατή απόσταση μεταξύ των επιστρεφόμενων γωνιών θέτοντας minDistance=3



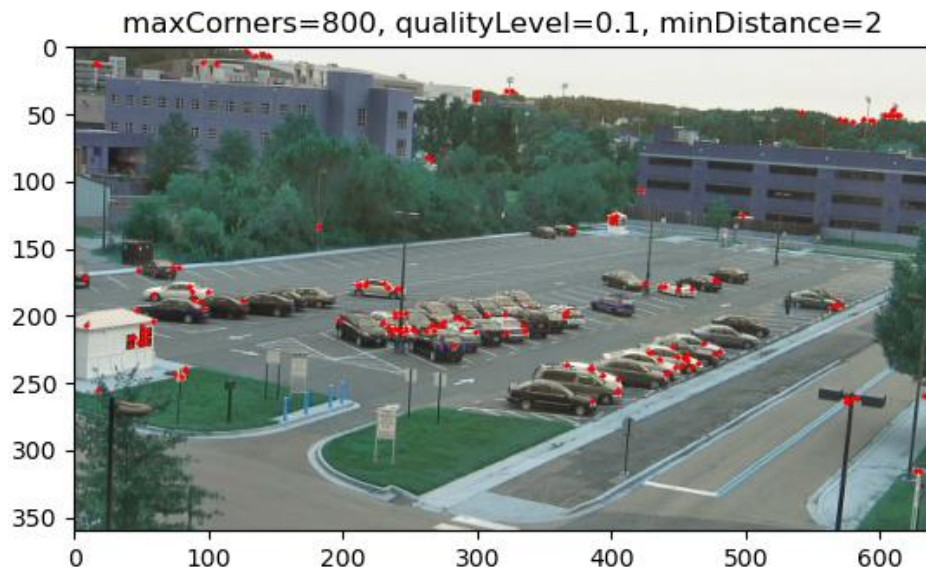
Για περαιτέρω βελτίωση του αποτελέσματος θέτουμε `minDistance=5` ώστε να κατανεμηθούν καλύτερα στο χώρο, οι γωνίες.

Harris corner

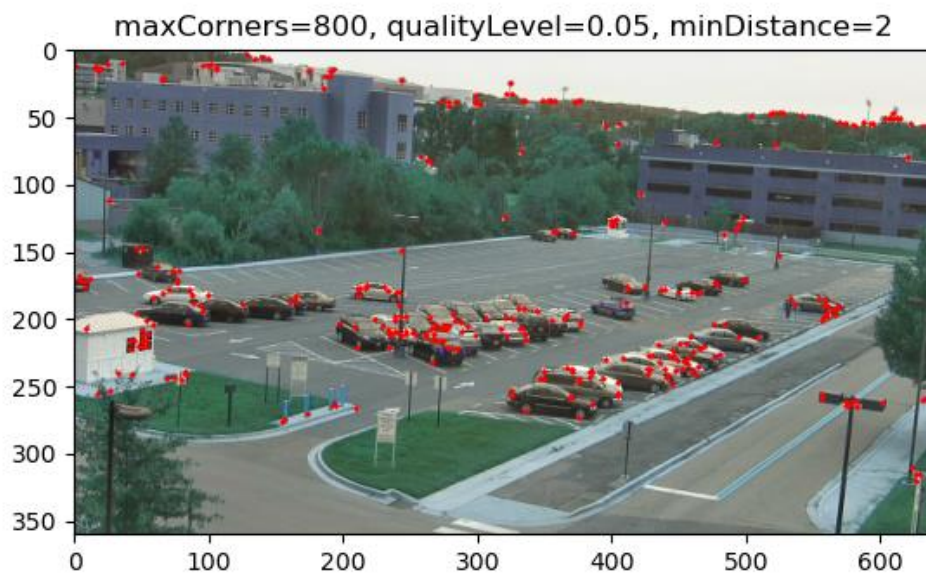
Στην συνάρτηση `cv2.goodFeaturesToTrack` θέτουμε `useHarrisDetector=True`



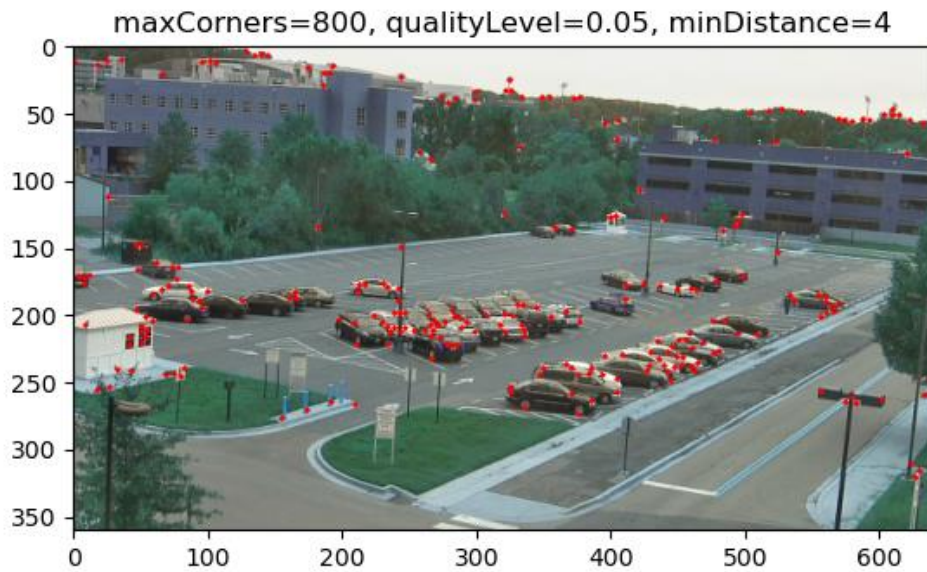
Παρατηρούμε ότι για τις συγκεκριμένες τιμές παραμέτρων αποτυπώνονται ελάχιστες γωνίες.



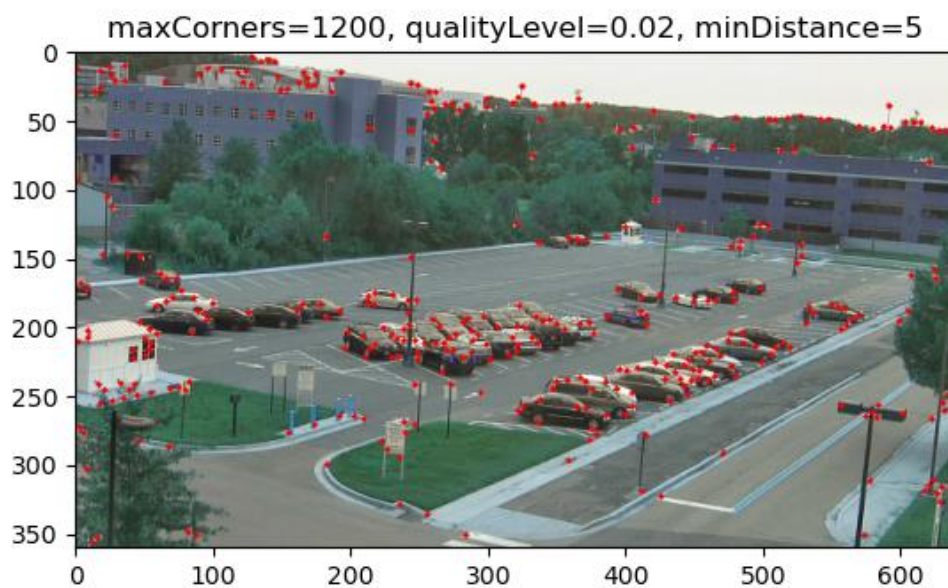
Αυξάνουμε τον αριθμό των επιστρεφόμενων γωνιών αλλά δεν παρατηρούμε σημαντικές μεταβολές .



Εν συνεχεία μεταβάλλουμε την τιμή του qualityLevel σε 0.05 και παρατηρούμε ότι αποτυπώνονται αρκετές γωνίες σε σχέση με τις δυο προηγούμενες περιπτώσεις.



Προκειμένου τα σημεία των γωνιών να διαμοιραστούν καλύτερα στο χώρο αυξάνουμε το `minDistance` και έτσι οδηγούμαστε σε καλύτερο αποτέλεσμα.



Τέλος αυξάνουμε ακόμη περισσότερο τον αριθμό των επιστρεφόμενων γωνιών, μειώνουμε λίγο ακόμη το `qualityLevel` και αυξάνουμε και `minDistance` ώστε να μην πέφτουν τα σημεία το ένα πάνω στο άλλο και οδηγούμαστε στο καλύτερο μέχρι στιγμής αποτέλεσμα.

- Στο [ερώτημα 4](#) εφαρμόζουμε τον αλγόριθμο Lucas-Kanade για τον υπολογισμό optical flow στα Harris και Shi-Tomasi σημεία του προηγούμενου ερωτήματος. Τα σχετικά αποτελέσματα παρουσιάζονται τρέχοντας τον κώδικα που παραδίδεται μαζί με την αναφορά. Κάποια ενδεικτικά screenshots που αποδεικνύουν ότι ο αλγόριθμος δουλεύει φαίνονται παρακάτω:

- Για σημεία που έχουν ανιχνευθεί με Shi-Tomasi corners detector

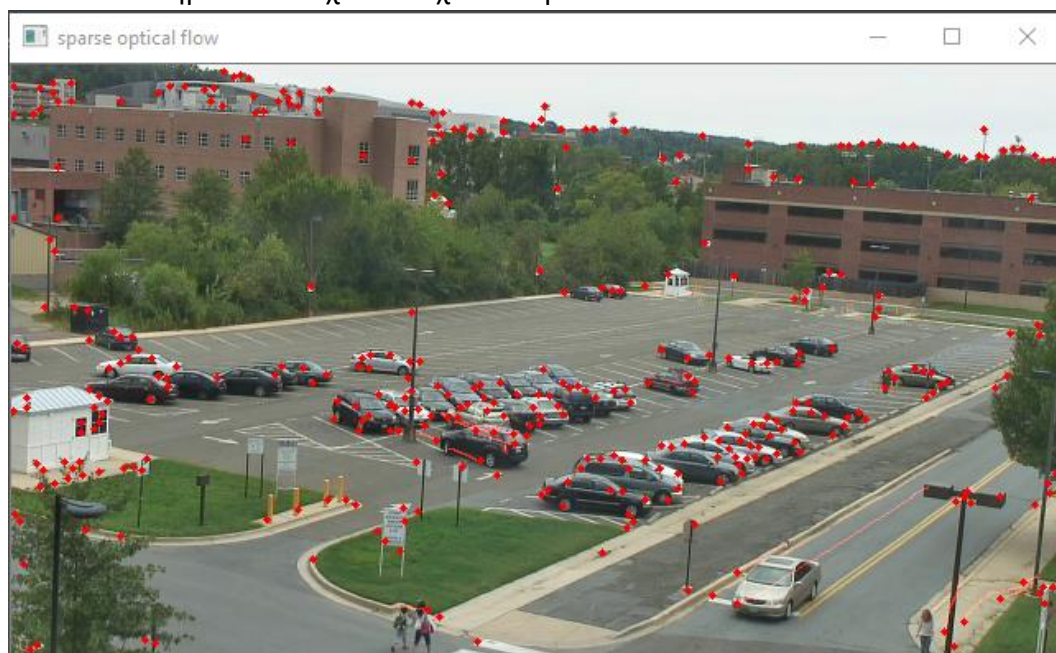


Παρατηρούμε ότι καταγράφηκε η πορεία του αυτοκινήτου που είχε ανιχνευθεί στο 1^ο frame και έφυγε από το παρκινγκ στην συνέχεια.



Καταγράφεται και η πορεία και άλλου αυτοκινήτου που μετακινείται σε σχέση με την θέση που είχε στο αρχικό frame. Άρα ο αλγόριθμος μας δουλεύει.

- Για σημεία που έχουν ανιχνευθεί με Harris corners detector





Παράλληλα να σημειώσουμε ότι όσον αφορά τις παραμέτρους της συνάρτησης `cv2.calcOpticalFlowPyrLK`, η οποία υπολογίζει την οπτική ροή, οι διαφορετικές τιμές στα κριτήρια δεν είχαν σημαντική διαφορά αλλά εμείς καταλήξαμε στον εξής συνδυασμό : **`criteria (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03)`**. Η κυριότερη παράμετρος, αυτή δηλαδή που επηρεάζει περισσότερο το οπτικό αποτέλεσμα, φαίνεται να είναι το **`windowSize`**. Εμείς επιλέξαμε **`windowSize=(15,15)`** ενώ για υψηλότερες ή χαμηλότερες τιμές εμφανίζονταν τυχαίες ή λανθασμένες γραμμές μεταξύ κορυφών ή δεν παρακολουθείται η κίνηση κάποιων αντικειμένων, όπως παρουσιάζεται παρακάτω:

- Για **`windowSize=(3,3)`**



- Για **windowSize=(25,25)**



- Στο [ερώτημα 5](#), τροποποιήσαμε τον κώδικα του προηγούμενου ερωτήματος, με στόχο την παρακολούθηση γωνιών Harris και Shi-Tomasi που εμφανίσθηκαν στο βίντεο μετά το πρώτο frame. Τα σχετικά αποτελέσματα παρουσιάζονται τρέχοντας τον κώδικα που παραδίδεται μαζί με την αναφορά. Κάποια ενδεικτικά screenshots που αποδεικνύουν ότι ο αλγόριθμος μας δουλεύει φαίνονται παρακάτω:

- Για σημεία που έχουν ανιχνευθεί με Shi-Tomasi corners detector



Παρατηρούμε ότι παρακολουθείται η πορεία του αυτοκινήτου στο δεξί μέρος της εικόνας, καθώς και των δυο ανθρώπων κάτω αριστερά, που αποτελούν στοιχεία που δεν είχαν εμφανιστεί στο αρχικό frame.



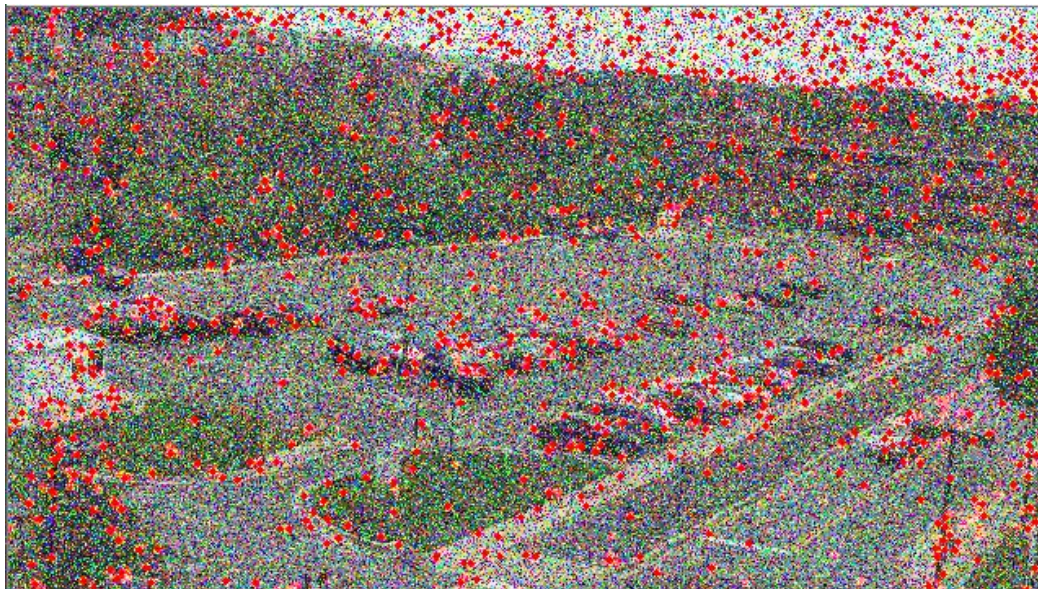


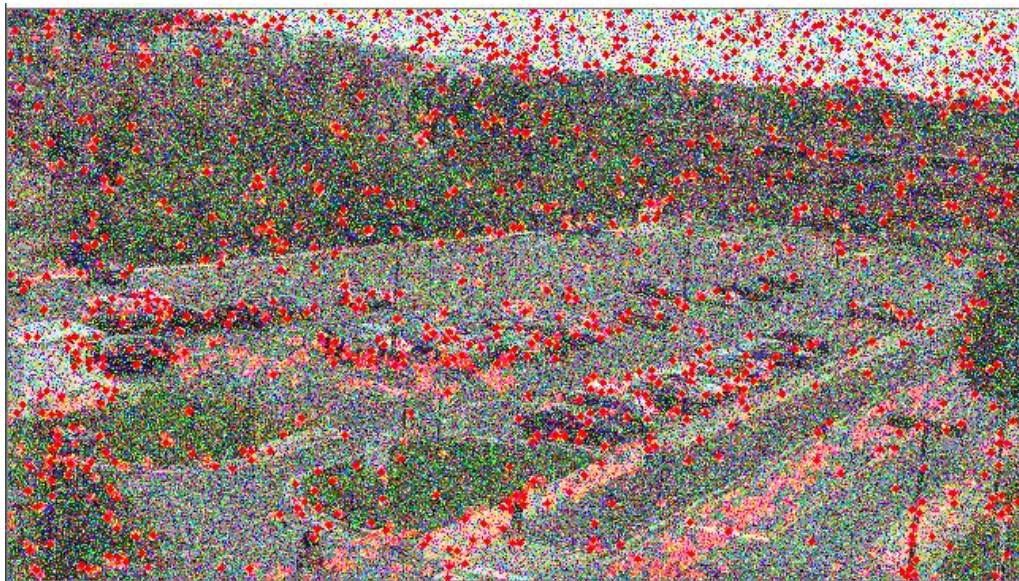
- Για σημεία που έχουν ανιχνευθεί με Harris corners detector





- Στο [ερώτημα 6](#), τροποποιήσαμε τον κώδικά σας για το βήμα 5, εισάγοντας Salt and Pepper θόρυβο μετά από κάθε "διάβασμα" ενός frame. Τα σχετικά αποτελέσματα παρουσιάζονται τρέχοντας τον κώδικα που παραδίδεται μαζί με την αναφορά. Κάποια ενδεικτικά screenshots που αποδεικνύουν ότι ο αλγόριθμος μας δουλεύει φαίνονται παρακάτω:



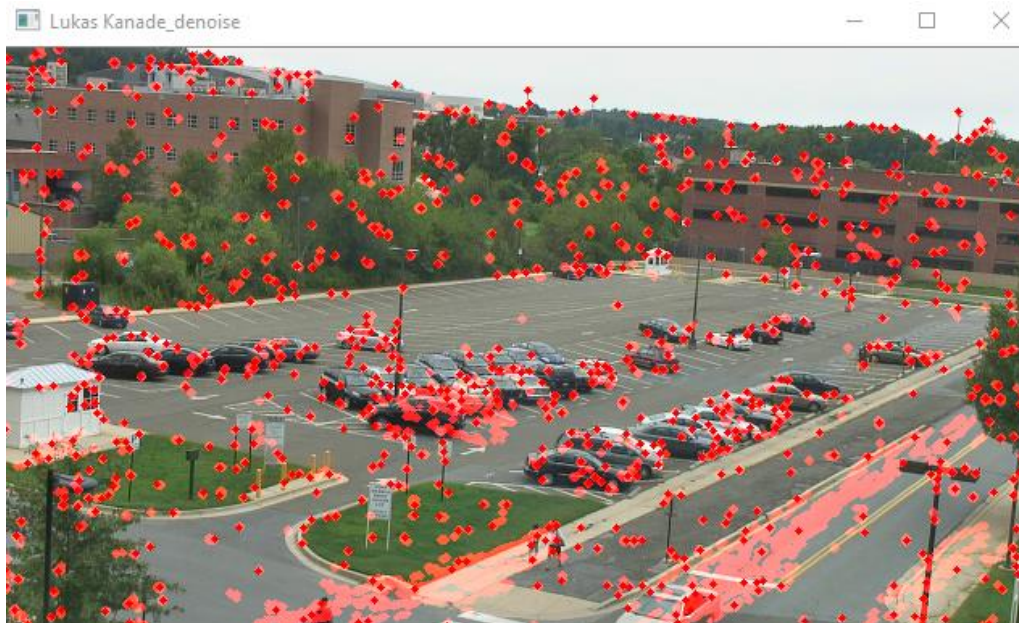


Παρατηρούμε ότι η εισαγωγή θορύβου δεν επιτρέπει την σωστή εφαρμογή του του αλγορίθμου Lukas Kanade, αφού σε αρκετά σημεία εντοπίζονται λανθασμένα corners. Ωστόσο μπορούμε να διακρίνουμε ότι και πάλι παρακολουθείται η ροή κάποιων αυτοκινήτων και ανθρώπων (δεν είναι ιδιαίτερα ευκρινές το αποτέλεσμα λόγω του θορύβου)

- Στο [ερώτημα 7](#) , εφαρμόσαμε κατάλληλη αποθορυβοποίηση, ώστε να ελαττώσουμε-αφαιρέσουμε τον θόρυβο που προσθέσαμε στο ερώτημα 6. Ωστόσο το αποτέλεσμα δεν είναι τόσο «σωστό» όσο το αντίστοιχο του βήματος 5, αφού υπάρχουν κάποιες random γωνίες. Τα σχετικά αποτελέσματα

παρουσιάζονται τρέχοντας τον κώδικα που παραδίδεται μαζί με την αναφορά.
Κάποια ενδεικτικά screenshots φαίνονται παρακάτω:

- Για **neighborhood = disk(radius=2)**



- Για **neighborhood = disk(radius=7)**

