

Exercise 1

1.

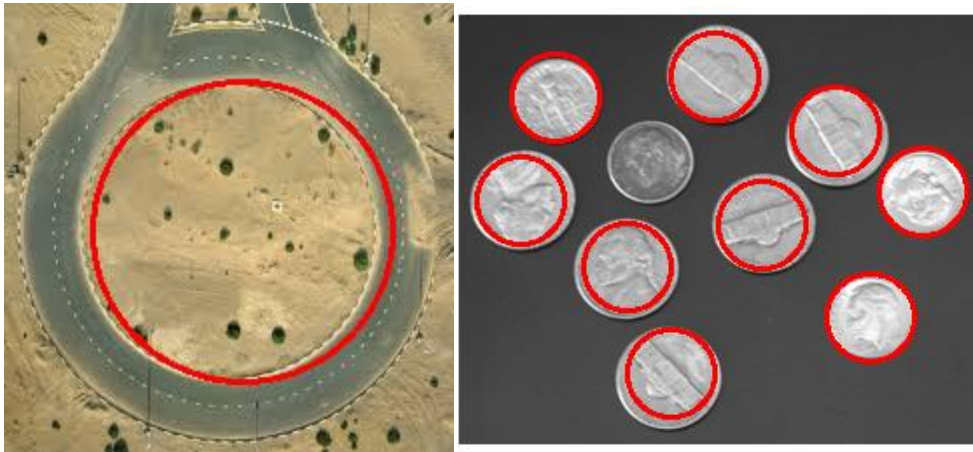
Υλοποιούμε τη συνάρτηση `BlobDedector(img, radius, thres_ratio=0.9)` που δέχεται σαν όρισμα μια εικόνα στην οποία θα γίνει αναζήτηση blobs, την ακτίνα που καθορίζει το μέγεθος των blobs που αναζητούμε και ένα ποσοστό (φωτεινότητας επί του κατ' απόλυτη τιμή φωτεινότερου pixel) που θα χρησιμοποιηθεί ως κατώφλι που θα καθορίζει την ευαισθησία (πόσο αυστηρό ή όχι θα είναι το κριτήριο) του εντοπισμού των blobs.

Η εικόνα αρχικά μετατρέπεται σε grayscale και βάσει της ακτίνας προσδιορίζεται το σ (sigma) για την δημιουργία του LoG φίλτρου που θα εφαρμοστεί σ' αυτήν.

Το αποτέλεσμα κατωφλιώνεται με βάση το ποσοστό που ορίσαμε στην συνάρτηση και καταλήγουμε (χρησιμοποιώντας έναν ενδιάμεσο πίνακα με Boolean τιμές ως 'μάσκα' ο οποίος συνδυάζεται με τον προηγούμενο πίνακα για την απομόνωση των Pixel εκείνων που οι τιμές τους ικανοποιούν το κριτήριο) στην δημιουργία εικόνας με τα υποψήφια για δημιουργία blobs pixel.

Από την τελευταία αυτή εικόνα μέσω της συνάρτησης `local_maxima_3D` (και αφού έχουμε προσθέσει μια επιπλέον διάσταση στην εικόνα μας για να τη μετατρέψουμε σε 3D) βρίσκουμε τα σημεία που αποτελούν τα κέντρα των blobs που με την προηγούμενη διαδικασία προσδιορίσαμε.

Έχοντας προσδιορίσει τα κέντρα των blobs και γνωρίζοντας την ακτίνα τους τα σχεδιάζουμε στην εικόνα.



Διαπιστώνουμε ότι ανάλογα με την επιλογή του κατωφλίου (όσο αυξάνεται τόσο πιο αυστηρό γίνεται το φίλτρο μας) έχουμε τη δυνατότητα σε σχέση με συγκεκριμένη ακτίνα να εντοπίζουμε πιο ξεκάθαρα ή λιγότερο ξεκάθαρα blobs με κίνδυνο (μειώνοντας αρκετά το κατώφλι) να εντοπίζονται (θεωρούνται) blobs ακόμη κι εκεί που για εμάς δεν υπάρχουν.

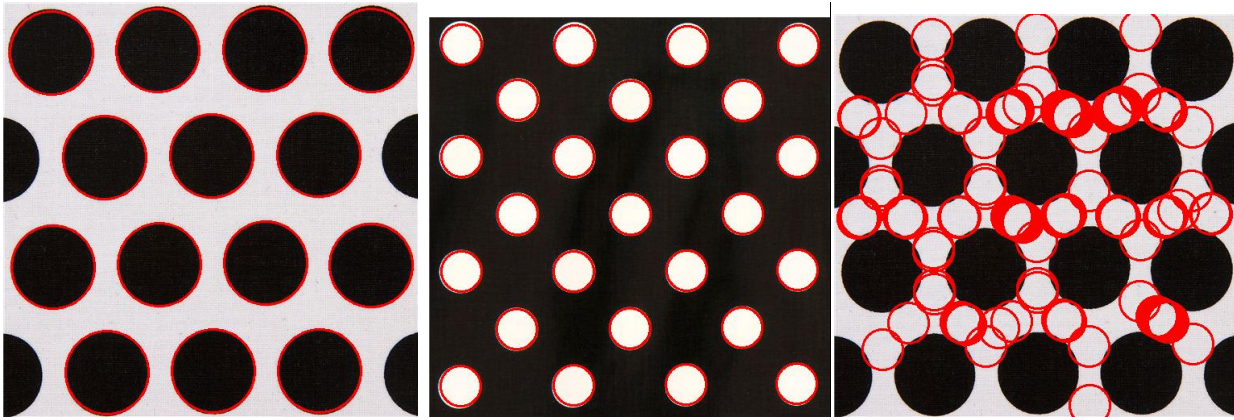
2.

Δημιουργούμε (σε συνέχεια της προηγούμενης) μια νέα συνάρτηση `BlobDedector_new(img, radius, thres_ratio=0.9, br=3)` η οποία δέχεται σαν όρισμα μια επιλογή για το εάν θα αναζητήσει τα περισσότερα σκοτεινά ή φωτεινά blobs ή θα αναζητήσει και τους δυο τύπους blobs.

Ακλουθούμε την ίδια με πριν διαδικασία μόνο που στην κατωφλίωση, βάσει της παραπάνω επιλογής, απομονώνουμε τα αντίστοιχα blobs.

Δεδομένου του τρόπου αναπαράστασης (φωτεινών – σκοτεινών pixel) της εικόνας που προέκυψε από την εφαρμογή του LoG φίλτρου επιλεγούμε τα pixel με μεγαλύτερη από το κατώφλι τιμή για τα σκουρόχρωμα blobs ή αυτά με την μικρότερη από την αντίθετη του κατωφλίου τιμή για τα ανοιχτόχρωμα ενώ αν θέλουμε και των δυο τύπων blobs συγκρίνουμε τις απόλυτες τιμές των pixel σε σχέση με το κριτήριο.

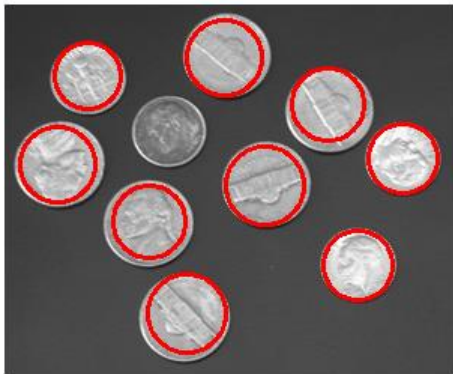
Παρατηρούμε ότι στοχευμένη αναζήτηση συνδυασμού φωτεινότητας, ακτίνας & κατωφλίου δίνουν αρκετά ικανοποιητικά αποτελέσματα στον εντοπισμό ανάλογων blobs ενώ ο detector παραπλανάται όταν π.χ. αναζητήσουμε αντίθετης φωτεινότητας από τα υπαρκτά blobs με χαμηλές απαιτήσεις κατωφλίωσης.

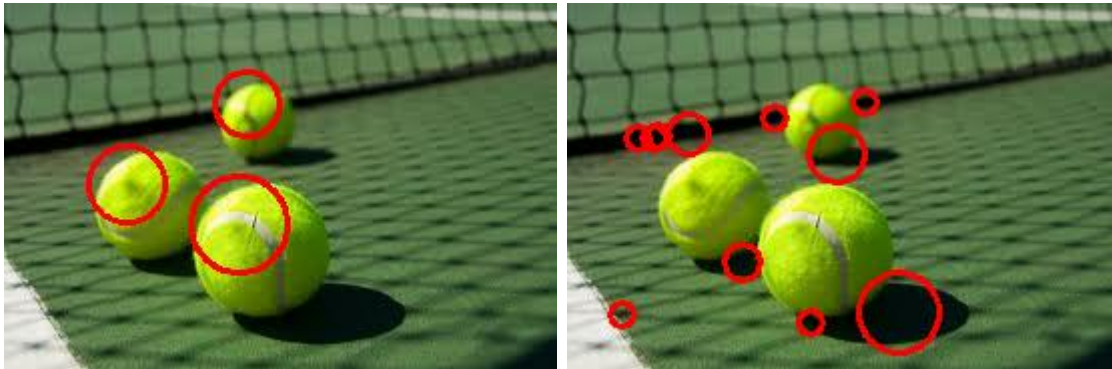
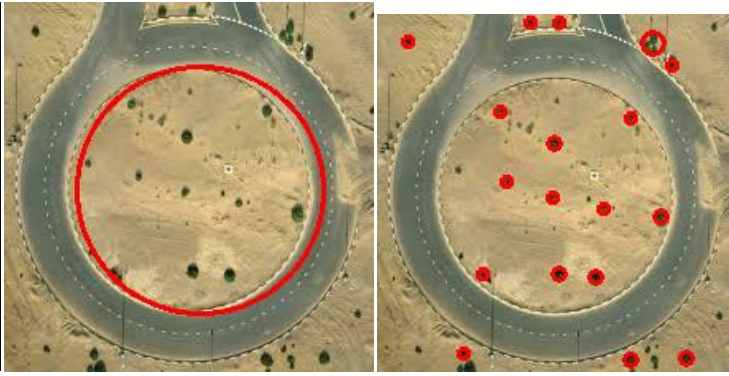


3.

Επεκτείνουμε τον ανιχνευτή `BlobDedector_extended(img, rad_min, rad_max, thres_ratio=0.9, br=3)` ώστε να δέχεται σαν επιπλέον όρισμα το εύρος ακτίνων των Blobs που πρόκειται να αναζητήσει και με την πληροφορία αυτή εκτελούμε ένα βρόγχο όπου βάσει της εκάστοτε τιμής ακτίνας εφαρμόζουμε αντίστοιχο LoG φίλτρο και αφού καταφλιώσουμε, ανάλογα με την επιλογή αναζητούμενης φωτεινότητας blobs, το αποτέλεσμα το τοποθετούμε σε ένα τρισδιάστατο πλέον πίνακα όπου εφαρμόζοντας την συνάρτησης `local_maxima_3D` βρίσκουμε τα σημεία που αποτελούν τα κέντρα των blobs που με την προηγούμενη διαδικασία προσδιορίσαμε.

Έχοντας προσδιορίσει τα κέντρα των blobs και γνωρίζοντας (από την πληροφορία της τρίτης διάστασης του εξαγόμενου από την προηγούμενη συνάρτηση πίνακα συντεταγμένων) από ποιο στη σειρά πέρασμα του βρόγχου προέκυψε αυτό το κέντρο, μπορούμε να προσδιορίσουμε την ακτίνα (βάσει της οποίας υπολογίστηκε το σ του LoG φίλτρου που εφαρμόσαμε στο συγκεκριμένο πέρασμα) του αντίστοιχου blob και να το σχεδιάσουμε στην εικόνα.





Σε συνέχεια των διαπιστώσεων μας από το προηγούμενο ερώτημα, παρατηρούμε ότι μεταβάλλοντας την επιλογή φωτεινότητας και το εύρος ακτίνας των αναζητούμενων blobs καθώς και την αυστηρότητα του κριτηρίου αποδοχής blobs τα αποτελέσματα διαμορφώνονται αντιστοίχως.

Exercise 2

Αρχικά υλοποιούμε (με τις συναρτήσεις `solve_affine(points1, points2)` & `solve_homography(points1, points2)` αντιστοίχα) τον Affine Μετασχηματισμό καθώς και τον Μετασχηματισμό Ομογραφίας (επίλυση με Linear Least Squares).

Εφαρμόζουμε, για δόκιμη, τους μετασχηματισμούς αυτούς μεταξύ γνωστών σημείων και διαπιστώνουμε ήδη από αυτά ότι ο Affine δυσκολεύεται να παρακολουθήσει την μη ομοιόμορφη μετατόπιση του συνόλου των σημείων ('στρέβλωση' / παραμόρφωση της σχετικής τους θέσης, έχουν αλλάξει οι σχετικές θέσεις των σημείων και οι ευθείες πλέον δεν είναι παράλληλες) ενώ ο μετασχηματισμός ομογραφίας μπορεί να υπολογίσει και τοποθετήσει σωστά τα σημεία αυτά.

Στη συνέχεια δοκιμάζουμε τους δυο μετασχηματισμούς για να προσδιορίσουμε τη θέση ενός αντικειμένου σε μια εικόνα αφού έχουμε προσδιορίσει και εξάγει ένα πλαίσιο του αντικειμένου αυτού από μια άλλη αρχική γνωστή εικόνα.

Για το σκοπό αυτό εντοπίζουμε χαρακτηριστικά του αντικειμένου στο πλαίσιο και χαρακτηριστικά από την εικόνα (στην οποία θέλουμε να προσδιορίσουμε τη θέση του) κι αφού βρούμε ποια ταιριάζουν επιλέγουμε (με βαθμό αυστηρότητας που καθορίζουμε) αυτά τα οποία θεωρούμε ότι ταιριάζουν καλύτερα.

Βάσει των σημείων που πρόεκυψαν από την παραπάνω διαδικασία υπολογίζουμε τον πίνακα για τον Αφινικό Μετασχηματισμό ή Ομογραφίας και τον εφαρμόζουμε στα σημεία του περιγράμματος του πλαισίου (του αντικειμένου που αναζητούμε) ώστε να υπολογίσουμε την θέση του πλαισίου στην εικόνα μας, η οποία θέση θα προσδιορίζει το αναζητούμενο αντικείμενο.

Και πάλι από το μετασχηματισμό αυτό διαπιστώνουμε ότι σε εικόνα που παρατηρούνται 'στρεβλώσεις' - παραβολικά φαινόμενα ακόμα κι αν έχουμε επαρκή πλήθος καλών και σωστών αντιστοιχίσεων ο Αφινικός μετασχηματισμός δεν κατορθώνει να τοποθετήσει με ακρίβεια το περίγραμμα στο αναζητούμενο αντικείμενο της εξεταζόμενης εικόνας ενώ ο μετασχηματισμός ομογραφίας εξασφαλίζει καλύτερα αποτελέσματα.

Στη συνέχεια, αφού εξάγουμε ένα πλαίσιο ενός αντικειμένου από μια αρχική εικόνα (όπου γνωρίζουμε τη θέση του μέσα αυτήν) προσδιορίζουμε τα χαρακτηριστικά του και με ένα βρόγχο διατρέχουμε μια σειρά από εικόνες στις

οποίες απεικονίζεται το αντικείμενο αυτό (το οποίο θα αναζητήσουμε), προσδιορίζουμε τα χαρακτηριστικά της εικόνας κάθε φορά και αφού εξετάσουμε ποια ταιριάζουν μ' αυτά του πλαισίου μας επιλέγουμε (με κριτήριο που διαμορφώνουμε) αυτά που θεωρούνται καλύτερα και με βάση αυτά υπολογίζουμε είτε τον Αφινικό μετασχηματισμό ή το μετασχηματισμό ομογραφίας (το ίδιο είδος μετασχηματισμού κάθε φορά για όλο το πλήθος των εικόνων). Εφαρμόζουμε τον επιλεγμένο μετασχηματισμό στα σημεία του περιγράμματος του αρχικού πλαισίου τοποθετώντας το έτσι στην εξεταζόμενη κάθε φορά εικόνα, θεωρώντας ότι το τοποθετούμενο πλαίσιο προσδιορίζει την θέση του αναζητούμενου αντικειμένου στην εικόνα αυτή.

Από τα αποτελέσματα, πέρα από τη προηγούμενη διαπίστωση σχετικά με τον εντοπισμό του αντικειμένου όταν έχουμε παραβολικά φαινόμενα (κίνηση κάμερας, στρεβλώσεις / παραμορφώσεις, αλλοιώσεις προβολής) παρατηρούμε ότι σε εικόνες, που λόγω και των φαινομένων αυτών, δεν έχουμε αρκετές καλές αντιστοιχίσεις ώστε να μπορέσουν να επιλυθούν οι παραπάνω μετασχηματισμοί αναγκάζομαστε να μειώσουμε τις απαιτήσεις μας για τον χαρακτηρισμό καλών αντιστοιχίσεων (μεγαλώνοντας την αποδεκτή απόσταση μεταξύ των εξεταζόμενων αντιστοιχιών) με αποτέλεσμα να αυξάνεται η πιθανότητα να θεωρούνται καλές και μη σωστές αντιστοιχίσεις οι οποίες όμως 'παραπλανούν' τους παραπάνω μετασχηματισμούς.

Έτσι παρατηρούμε ειδικά στον μετασχηματισμό ομογραφίας ότι σε κάποιες περιπτώσεις αδυνατεί να τοποθετήσει ή και να σχηματίσει το πλαίσιο ακόμα και σε εικόνες που με αυστηρή επιλογή χαρακτηριστικών τα κατάφερνε πολύ καλά - καλύτερα από τον Αφινικό. Επίσης παρατηρούμε ότι σ' αυτές τις περιπτώσεις ο Αφινικός μετασχηματισμός μπορεί καλύτερα να τοποθετήσει το πλαίσιο στο χώρο της εικόνας ακόμη κι αν η τοποθέτηση σε σχέση με τη πραγματική θέση του αντικειμένου δεν είναι ικανοποιητική. Διαπιστώνουμε δηλαδή ότι και οι δυο μετασχηματισμοί και μάλλον περισσότερο αυτός της ομογραφίας είναι ευαίσθητοι (επηρεάζονται δηλαδή και δεν υπολογίζονται σωστά) στην ύπαρξη μη σωστών αντιστοιχίσεων κι όσο περισσότερες είναι αυτές ή/και περισσότερο λάθος είναι, τόσο πιο εσφαλμένος ο υπολογισμός τους.

Στη συνέχεια και προκειμένου να ξεπεράσουμε το πρόβλημα αυτό των λάθος αντιστοιχίσεων που επηρεάζουν τον σωστό υπολογισμό των μετασχηματισμών μας, εφαρμόσαμε τον αλγόριθμο Ransac για την επιλογή κατάλληλου υποσύνολου σημείων, από αυτά που συνολικά πρόεκυψαν από την κατωφλίωση των ταιριασμάτων, που δίνουν την καλύτερη δυνατή (ανάλογα και με τις παραμέτρους εκτέλεσης του αλγορίθμου) προσέγγιση για σωστό υπολογισμό του ενός ή του άλλου μετασχηματισμού.

Η εφαρμογή του αλγορίθμου αυτού έγινε με τη συνάρτηση `My_ransac(tr, p1, p2, N, p_nub, dist, th=0.8)` η οποία δέχεται σαν όρισμα μια τιμή που καθορίζει ποιον μετασχηματισμό θα εφαρμόσει ο αλγόριθμος, τα σημεία (όπως αυτά πρόεκυψαν από την κατωφλίωση των ταιριασμάτων) του πλαισίου και τα αντίστοιχα τους στην εικόνα, τον μέγιστο αριθμό επαναλήψεων που θα πραγματοποιήσει (ώστε να βρει το καλύτερο μετασχηματισμό), το μέγεθος του υποσύνολου των σημείων που θα χρησιμοποιήσει, την απόσταση με βάση την οποία θα κρίνει εάν κάποιο από τα σημεία της εικόνας θεωρείται outlier ή inliner για τον τρέχον υπολογισμό του μετασχηματισμού και το ποσοστό επί του πλήθους των σημείων της εικόνας που εάν χαρακτηριστεί ως inliner θεωρούμε ότι ο υπολογισμός του μετασχηματισμού είναι ικανοποιητικός.

Η παραπάνω συνάρτηση επιλεγεί τυχαία κάθε φορά συγκεκριμένου πλήθους υποσύνολο του συνόλου σημείων του πλαισίου και των αντίστοιχων τους από την εικόνα, υπολογίζει (εάν το πλήθος που ορίζουμε είναι μικρότερο από το πλήθος ολόκληρου του συνόλου σημείων του πλαισίου αλλιώς εφαρμόζει μετασχηματισμό βάσει όλων των σημείων που πρόεκυψαν από την διαδικασία ταιριάσματος) τον μετασχηματισμό που έχει επιλεγεί, εφαρμόζει τον μετασχηματισμό στα σημεία της εικόνας, ελέγχει την απόσταση των υπολογισμένων από τα πραγματικά σημεία της εικόνας και βρίσκει τα inliner. Αν τα inliner που προκύπτουν από το τρέχον πέρασμα – μετασχηματισμό είναι περισσότερα από τα έως τότε περισσότερα κρατάει το μετασχηματισμό αυτόν ως επικρατέστερο. Η διαδικασία συνεχίζεται έως ότου προκύψει πλήθος inliner μεγαλύτερο ή ίσο με αυτό που ορίστηκε ως αποδεκτό ή ολοκληρωθεί ο μέγιστος αριθμός επαναλήψεων που επιλέχθηκε.

Η εφαρμογή του αλγορίθμου στην εξέταση της παραπάνω σειράς εικόνων για την αναζήτηση του αντικειμένου, εξαλείφει την επίδραση των μη σωστών ταιριασμάτων και ειδικά στην περίπτωση του συνδυασμού του με τον μετασχηματισμό ομογραφίας δίνει αρκετά ικανοποιητικά αποτελέσματα εντοπισμού – παρακολούθησης του αντικειμένου σ' αυτές.