Target Locking and Tracking Using ORB

Εισαγωγή

Το πρόβλημα του object tracking είναι ένα από τα πιο σημαντικά προβλήματα στον κλάδο της ψηφιακής επεξεργασίας εικόνας, καθώς μπορεί να εφαρμοστεί για στην υλοποίηση πληθώρας εφαρμογών, όπως στην καθοδήγηση αυτόνομων αυτοκινήτων, στην αλληλεπίδραση ανθρωπου-μηχανης, στη παρακολούθηση βίντεο κ.α.

Σε αυτή την αναφορά θα αναλύσουμε την επίδοση ενός προγράμματος object tracking, το οποίο για αναγνώριση αντικειμένων χρησιμοποιεί τον αλγόριθμο **ORB** (Oriented FAST Rotated BRIEF) της βιβλιοθήκης OpenCV, σε συνδυασμό με έναν αλγόριθμο συσταδοποίησης. Η ακρίβεια του προγράμματος αξιολογήθηκε, σε ένα βίντεο πτήσης drone, για την παρακολούθηση ενός αντικείμενου, με κριτήριο το ποσοστό του βίντεο όπου ο στόχος έχει εντοπιστεί σωστά.

ΣΗΜΕΙΩΣΗ :

Στην αναφορά αυτή, δεν γίνεται ανάλυση της απόδοσης του προγράμματος σε πολλαπλά βίντεο, άλλα μόνο σε ένα. Αυτό καθιστά την αναφορά αυτή ως αναφορά απόδειξης αρχής ορθότητας και όχι αναφορά πρωτοτύπου.

ORB

Ο αλγόριθμος ORB είναι ένας αλγόριθμος εξαγωγής χαρακτηριστικών εικόνας, ο

οποίος είναι αρκετά γρήγορος και αποτελεσματικός για να χρησιμοποιηθεί σε εφαρμογές πραγματικού χρόνου. Αναπτύχθηκε από τους Ethan Rublee, Vincent Rabaud, Kurt Konolige, και Gary R. Bradski το 2011 στα OpenCV labs ως open-source εναλλακτική του SIFT και του SURF, καθώς οι αλγόριθμοι αυτοί είναι πατενταρισμένοι και δεν μπορούν να χρησιμοποιηθούν χωρίς άδεια.

Ο ORB συνδυάζει δυο αλγορίθμους, τον FAST (Features from Accelerated Segment Test) για εξαγωγή χαρακτηριστικών και τον rBRIEF (Rotation-aware Binary Robust Indepenent Elementary Features) για την περιγραφή αντικειμένων από τα προηγουμένως εξαγόμενα χαρακτηριστικά.

Οι αλγόριθμοι αυτοί έχουν υποστεί μερικές μετατροπές έτσι ώστε να δρουν αποτελεσματικά σε εικόνες ανεξαρτήτως μεγέθους και περιστροφής. Για την ανεξαρτησία μεγέθους, εφαρμόζεται στην εικόνα μια Γκαουσιανή πυραμίδα, στο αποτέλεσμα της οποίας εφαρμόζεται μετά ο FAST, ενώ για την ανεξαρτησία περιστροφής, εφαρμόζεται το κεντροειδές έντασης φωτεινότητας.

Παρόλο που ο ORB χρησιμοποιεί τον FAST για εξαγωγή χαρακτηριστικών, είναι εφικτό να χρησιμοποιήσουμε διαφορετικούς αλγορίθμους. Στο πρόγραμμα μας χρησιμοποιούμε τον αλγόριθμο cv::goodFeaturesToTrack() ο οποίος χρησιμοποιεί τον Shi-Tomashi Corner Detector. Για την περιγραφή των χαρακτηριστικών εξακολουθούμε να

χρησιμοποιούμε τον rBRIEF του ORB.

Συσταδοποίηση

Αφού εφαρμοστεί ο ORB στην φωτογραφία του αντικειμένου που επιθυμούμε να ακολουθήσουμε και στο πηγαίο βίντεο του UAV, εφαρμόζεται ο αλγόριθμος Brute-Force Matcher με απόσταση NORM_HAMMING2 για την αντιστοίχηση κάθε χαρακτηριστικού από την μια εικόνα στην άλλη. Με την ολοκλήρωση της αντιστοίχησης, είναι πιθανό να μην έχουν αντιστοιχηθεί σωστά όλα τα χαρακτηριστικά. Για να ακολουθήσουμε λοιπόν το επιθυμητό αντικείμενο θα πρέπει να επιλέξουμε την περιοχή με τις περισσότερες αντιστοιχίσεις. Η επιλογή αυτή γίνεται μέσο της εφαρμογής ενός δικού μας αλγορίθμου συσταδοποίησης.

Ο αλγόριθμος αυτός ταξινομεί n χαρακτηριστικά (F με χαρακτηριστικό στη θέση i f_i) με βάση αρχικά την απόσταση τους από το κέντρο του άξονα (0,0). Στη συνέχεια συγκρίνει ένα ένα τα χαρακτηριστικά με βάση την ευκλείδεια απόσταση τους. Αν η απόσταση μεταξύ ενός χαρακτηριστικού f_i και του προηγούμενού του f_{i-1} είναι μικρότερη από d τότε προσθέτουμε το f_i στην συστάδα c_j. Αν η απόσταση είναι μεγαλύτερη από k τότε προσθέτουμε το χαρακτηριστικό σε νέα συστάδα c_{j+1}. Επαναλαμβάνουμε την παραπάνω διαδικασία μέχρι να έχουν τοποθετηθεί όλα τα χαρακτηριστικά σε μια συστάδα. Τέλος επιλέγουμε την συστάδα με τα περισσότερα χαρακτηριστικά ως εκείνη που περιγράφει τον πραγματικό στόχο, υπολογίζουμε το κεντροειδές της χρησιμοποιώντας τον μέσο όρο

των τοποθεσιών όλων των χαρακτηριστικών της, και χρησιμοποιούμε το κεντροειδές αυτό για την ταξινόμηση του επόμενου συνόλου χαρακτηριστικών (αντί για την αρχή του άξονα).

Με αυτό τον αλγόριθμο, επιτυγχάνουμε ταχεία τοπική συσταδοποιηση γύρο από ένα κεντροειδές ενδιαφέροντος σε διάμετρο d, με περιπλοκότητα $O(nlogn)$ για την ταξινόμηση + $O(n)$ για την σύγκριση των στοιχείων και την εύρεση της μέγιστης συστάδας = $O(nlogn)$.

Εφαρμογή

Για να αξιολογήσουμε την αποδοτικότητα του προγράμματος, το τρέχουμε σε ένα βίντεο πτήσης drone, όπου επιλέγουμε ως στόχο ένα δέντρο. Χρησιμοποιούμε δυο διαφορετικές φωτογραφίες στόχου, σε διαφορετικές αποστάσεις (μια κοντά και μια μακριά), αλλά και διαφορετικές παραμέτρους για τον ORB και τον αλγόριθμο συσταδοποίησης. Συγκεκριμένα δοκιμάζουμε διαφορετικές τιμές για τις παραμέτρους WTA_K

1: Μακρινή φωτογραφία (αριστερά) και κοντινή φωτογραφία (δεξιά)

και edgeThreshold, για τον ORB, ενώ για τον αλγόριθμο συσταδοποιησης δοκιμάζουμε διαφορετικές τιμές για το d (διάμετρο περιοχής ενδιαφέροντος).

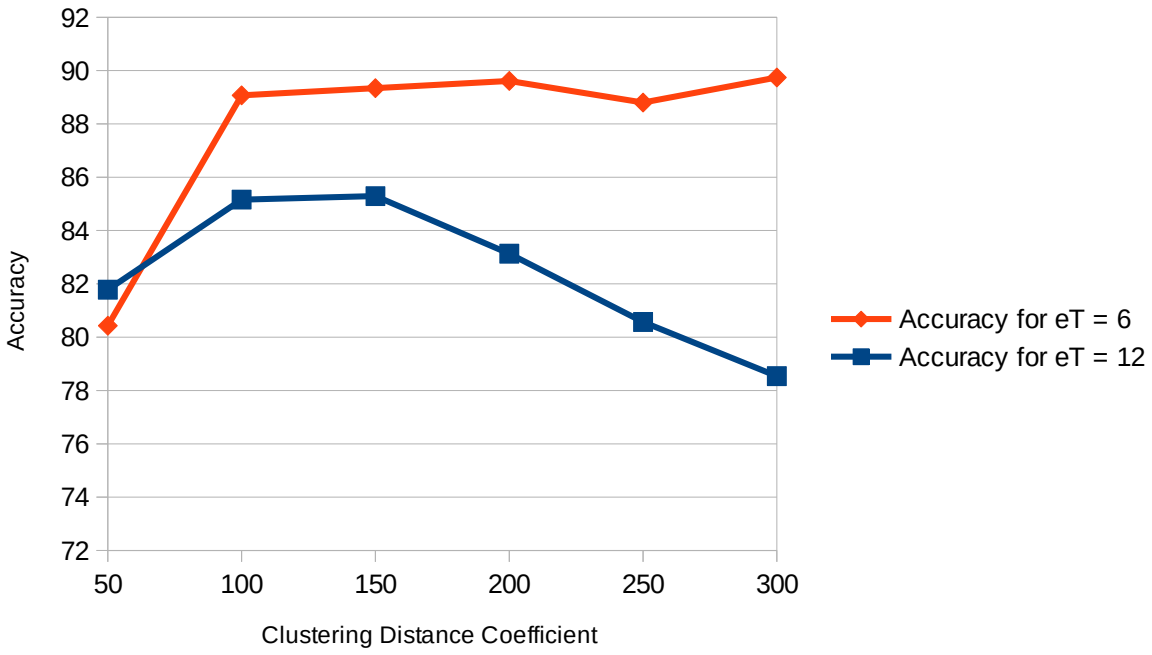
Στις φωτογραφίες 2 και 3, διακρίνονται με κόκκινο χρώμα,

2: Στόχος : μακρινός | WTA_K = 2 | edgeThreshold = 6 | d = 50 | Πρόβλεψη : Λάθος

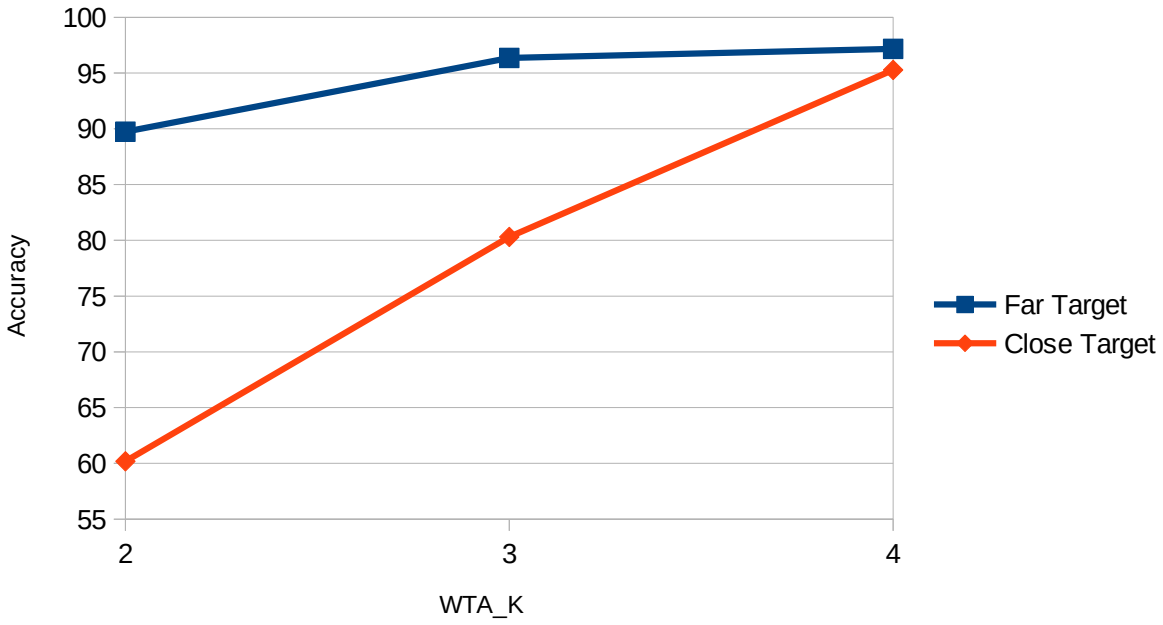
3: Στόχος : μακρινός | WTA_K = 2 | edgeThreshold = 6 | d = 150 | Πρόβλεψη : Σωστή

τα χαρακτηριστικά που έχουν επιλεχθεί από τον αλγόριθμο συσταδοποιησης, ενώ τα υπόλοιπα χαρακτηριστικά εμφανίζονται ως λευκά.

Από την παραπάνω φωτογραφία φαίνεται πως όταν τα χαρακτηριστικά είναι πιο



Διάγραμμα 4



Διάγραμμα 5

αραιά, το πρόγραμμα επιλέγει λάθος περιοχή ως περιοχή ενδιαφέροντος. Για την διόρθωση του προβλήματος αυτού, αρκεί να αυξήσουμε την παράμετρο d για τον αλγόριθμο συσταδοποίησης.

Στο διάγραμμα 4 παρατηρούμε πως για WTA_K = 2 στην μακρινή φωτογραφία στόχου, αν αυξήσουμε την παράμετρο απόστασης για τον αλγόριθμο συσταδοποίησης, το πρόγραμμα αποδίδει καλύτερα. Βέβαια μπορούμε επιπλέον να διακρίνουμε πως το πρόγραμμα είναι πιο ακριβές όταν το edgeTreshold είναι ίσο με 6 αντί για 12, και ενώ παρουσιάζει βελτίωση με την αύξηση της παραμέτρου απόστασης, μετά την τιμή των 150 pixel η ακρίβεια φθίνει.

Ένας άλλος τρόπος να βελτιώσουμε την ακρίβεια του προγράμματος, είναι να αυξήσουμε την τιμή της WTA_K, η οποία καθορίζει τον αριθμό των σημείων που παράγουν κάθε rBRIEF descriptor. Αυξάνοντας την παράμετρο αυτή, παρατηρούμε αυξημένη ακρίβεια χωρίς να επηρεάζεται ο χρόνος εκτέλεσης. Στο διάγραμμα 5 παρουσιάζουμε την αυτή τη βελτίωση, καθώς και την διαφορά στην ακρίβεια όταν χρησιμοποιούμε φωτογραφίες στόχου από διαφορετικές αποστάσεις. Να σημειωθεί πως η τιμή του edgeTreshold είναι ίση με 6 και για την παράμετρο d του αλγορίθμου συσταδοποίησης έχουν επιλεχθεί οι τιμές που απέδωσαν την μεγαλύτερη ακρίβεια.

Παρατηρούμε λοιπόν πως με σωστό ορισμό των παραμέτρων, το πρόγραμμα μπορεί να εντοπίσει τον στόχο με μέχρι και 97.17% επιτυχία. Επιπλέον παρατηρήθηκε σχεδόν διπλάσιος χρόνος

επεξεργασίας για κάθε καρέ βίντεο όταν χρησιμοποιήθηκε η κοντινή φωτογραφία του στόχου. Το γεγονός αυτό, σε συνδυασμό με τα χαμηλότερα ποσοστά ακρίβειας, μας οδηγούν στο συμπέρασμα πως δεν είναι βέλτιστο να χρησιμοποιούμε για στόχους, φωτογραφίες με ανάλυση κοντά στην ανάλυση του βίντεο.

Συμπέρασμα

Συμπεραίνουμε λοιπών πως το πρόγραμμα καταφέρνει, με μεγαλύτερη από 90% ακρίβεια, να εντοπίσει σε πραγματικό χρόνο ένα στόχο από μια φωτογραφία. Να σημειωθεί βέβαια πως το πρόγραμμα εξετάστηκε μόνο σε ένα βίντεο. Όπως αναφέραμε στην εισαγωγή η αναφορά αυτή εκτελεί τον σκοπό της ως αναφορά απόδειξης ορθότητας αρχής του προγράμματος και όχι αναφορά πρωτότυπου. Πιθανότατα σε βίντεο διαφορετικής ανάλυσης, με διαφορετικά μεγέθη στόχων, οι παράμετροι να πρέπει να είναι διαφορετικοί για να παραχθεί το βέλτιστο αποτέλεσμα.

Πηγές

- Introduction to ORB (Oriented FAST and Rotated BRIEF)

  https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

- Shi-Tomasi Corner Detector & Good Features to Track

  https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html