
Ψηφιακή Επεξεργασία Εικόνas

Εργασία 2: Ανίχνευση ακμών και κύκλων

[Άνοιξη 2025]

1. Εισαγωγή

Η παρούσα εργασία επικεντρώνεται στην υλοποίηση και αξιολόγηση βασικών τεχνικών ανίχνευσης ακμών και κυκλικών σχημάτων σε ψηφιακές εικόνες, με χρήση της γλώσσας προγραμματισμού Python. Στόχος είναι η εξοικείωση με θεμελιώδεις μεθόδους επεξεργασίας εικόνas, όπως οι τελεστές Sobel και Laplacian of Gaussian (LoG), καθώς και η εφαρμογή του μετασχηματισμού Hough για την αναγνώριση κυκλικών αντικειμένων.

Πιο συγκεκριμένα, υλοποιήθηκαν και μελετήθηκαν:

- Ο τελεστής Sobel
- Ο τελεστής Laplacian of Gaussian (LoG)
- Η μέθοδος Hough

Η εργασία περιλαμβάνει την περιγραφή και ερμηνεία των υλοποιήσεων, την εφαρμογή τους σε συγκεκριμένη εικόνα με στόχο τον εντοπισμό μίας μπάλας, καθώς και την οπτική αξιολόγηση των αποτελεσμάτων σε διαφορετικές παραμετροποιήσεις.

Όλες οι συναρτήσεις έχουν υλοποιηθεί με τρόπο ώστε να υποστηρίζουν παραμετροποιημένη χρήση και σε άλλες εικόνες, πέραν της δοθείσας, εφόσον πληρούνται οι θεωρητικές προϋποθέσεις των μεθόδων.

2. Περιγραφή λειτουργίας των συναρτήσεων

Στην παρούσα ενότητα περιγράφεται η θεωρητική λειτουργία των βασικών συναρτήσεων που υλοποιήθηκαν στο πλαίσιο της εργασίας. Κάθε συνάρτηση σχεδιάστηκε ώστε να εξυπηρετεί ένα επιμέρους βήμα της διαδικασίας ανίχνευσης ακμών ή κυκλικών σχημάτων σε εικόνες, όπως αυτά περιγράφονται αναλυτικά στην εκφώνηση και στη σχετική θεωρία.

Υλοποιήθηκαν αποκλειστικά οι συναρτήσεις που ζητούνται ρητά στην εκφώνηση:

- `fir_conv()`
- `sobel_edge()`
- `log_edge()`
- `circ_hough()`

Ακολουθεί αναλυτική περιγραφή της θεωρητικής λειτουργίας κάθε συνάρτησης, οργανωμένη ανά αρχείο Python:

2.1 Αρχείο `fir_conv.py`

- `fir_conv(in_img_array: np.ndarray, h: np.ndarray, in_origin: np.ndarray = None, mask_origin: np.ndarray = None) -> tuple[np.ndarray, np.ndarray]:`

Η συνάρτηση `fir_conv()` υλοποιεί τη **συνέλιξη** (convolution) μιας **εικόνας** με μια **συνελικτική μάσκα FIR** (Finite Impulse Response), με χρήση μηδενικής επέκτασης (zero-padding). Η υλοποίηση στηρίζεται στην εφαρμογή της διακριτής συνέλιξης όπως περιγράφεται θεωρητικά στην εκφώνηση της εργασίας.

Αρχικά, επεκτείνεται η εικόνα με κατάλληλο zero-padding, ώστε να μπορεί η μάσκα να "σαρώσει" ολόκληρη την περιοχή. Στη συνέχεια, η μάσκα αντιστρέφεται και στους δύο άξονες (flip) και εφαρμόζεται μέσω διαδοχικού υπολογισμού εσωτερικού γινομένου σημείο προς σημείο (dot product) σε κάθε περιοχή της εικόνας. Το αποτέλεσμα είναι μια εικόνα εξόδου μεγαλύτερων διαστάσεων, στην οποία έχουν ληφθεί υπόψη τα όρια συνέλιξης.

Τελικά, επιστρέφει τόσο την εικόνα εξόδου όσο και την νέα θέση της αρχής στην έξοδο, η οποία προκύπτει ως άθροισμα των δύο αρχών (εισόδου και μάσκας). Η θέση αυτή είναι σημαντική για την κατάλληλη ευθυγράμμιση στις επόμενες φάσεις επεξεργασίας.

Η `fir_conv` αποτελεί βασικό υπολογιστικό εργαλείο για την εφαρμογή τελεστών όπως οι Sobel και LoG, και παρέχει τη γενικευμένη υποδομή συνέλιξης που απαιτείται από την εκφώνηση

2.2 Αρχείο *sobel_edge.py*

- *sobel_edge(in_img_array: np.ndarray, thresh: float) -> np.ndarray:*

Η συνάρτηση *sobel_edge()* υλοποιεί την ανίχνευση ακμών σε grayscale εικόνες με τη βοήθεια του **τελεστή Sobel**, ο οποίος υπολογίζει προσεγγιστικά τις παραγώγους πρώτης τάξης της εικόνας κατά τις δύο κατευθύνσεις της, παρέχοντας το μέτρο της τοπικής κλίσης που χρησιμοποιείται για τον εντοπισμό περιοχών με έντονες μεταβολές φωτεινότητας.

Η συνάρτηση εφαρμόζει δύο διακριτές μάσκες Sobel διαστάσεων 3×3 : μία κατά την οριζόντια διεύθυνση (G_x) και μία κατά την κατακόρυφη (G_y), οι οποίες περιγράφονται και θεωρητικά στην εκφώνηση. Οι μάσκες εφαρμόζονται στην εικόνα εισόδου μέσω της συνάρτησης *fir_conv*, και προκύπτουν δύο πίνακες που αντιστοιχούν στο τοπικό gradient σε κάθε διεύθυνση.

Το συνολικό μέτρο της κλίσης (gradient magnitude) σε κάθε σημείο υπολογίζεται με τη σχέση:

$$g(n_1, n_2) = \sqrt{g_1(n_1, n_2)^2 + g_2(n_1, n_2)^2}$$

όπως αναφέρεται και στη θεωρία.

Στη συνέχεια εφαρμόζεται ένα κατώφλι (threshold), ώστε να εντοπίζονται μόνο τα σημεία της εικόνας όπου το μέτρο της κλίσης υπερβαίνει την προκαθορισμένη τιμή. Το τελικό αποτέλεσμα είναι μια δυαδική εικόνα στην οποία τα σημεία με έντονες μεταβολές έντασης —δηλαδή οι ακμές— έχουν τιμή 1, και όλα τα υπόλοιπα έχουν τιμή 0.

Η συνάρτηση υποστηρίζει παραμετρικό καθορισμό του κατωφλίου, προσφέροντας ευελιξία στη ρύθμιση της ευαισθησίας της ανίχνευσης.

2.3 Αρχείο *log_edge.py*

- *log_edge(in_img_array: np.ndarray, epsilon: float, pattern: int) -> np.ndarray:*

Η συνάρτηση *log_edge()* υλοποιεί την ανίχνευση ακμών σε grayscale εικόνες μέσω του **τελεστή Laplacian of Gaussian (LoG)**, βασισμένη στην αρχή των **μηδενικών διελεύσεων (zero-crossings)** της δεύτερης παραγώγου. Ο LoG τελεστής συνδυάζει δύο στάδια: εξομάλυνση (Gaussian smoothing) της εικόνας για αποθρομβοποίηση, και συνέλιξη με διακριτή μάσκα Laplacian για τον εντοπισμό περιοχών όπου η δεύτερη παράγωγος της

έντασης της εικόνας μεταβάλλεται απότομα. Η συνέλιξη πραγματοποιείται με χρήση της συνάρτησης *fir_conv()*, και η μάσκα LoG έχει διαστάσεις 5×5 με κεντρική ενίσχυση (τιμή +16) και περιφερειακή εξασθένηση.

Η βασική ιδέα της μεθόδου είναι ότι οι ακμές αντιστοιχούν σε τοπικές αλλαγές πρόσημου στο αποτέλεσμα της δεύτερης παραγώγου (δηλαδή στο αποτέλεσμα της συνέλιξης). Το *log_edge()* εντοπίζει αυτές τις αλλαγές χρησιμοποιώντας τρεις εναλλακτικούς τρόπους, ελεγχόμενους από την παράμετρο *pattern*. Οι τρεις διακριτές στρατηγικές εντοπισμού zero-crossings έχουν διαφορετική αυστηρότητα και εφαρμόζονται για διαφορετικές απαιτήσεις ευαισθησίας και αξιοπιστίας στην ανίχνευση. Αναλυτικότερα:

- *pattern* = 1 → Απλή ανίχνευση σε 3×3 γειτονιά
Εντοπίζονται όλα τα pixels που παρουσιάζουν αλλαγή πρόσημου (zero-crossing) στο εσωτερικό της τοπικής 3×3 γειτονιάς, σε οποιαδήποτε κατεύθυνση. Η μέθοδος είναι ιδιαίτερα ευαίσθητη αλλά ενδέχεται να παράγει ψευδείς ανιχνεύσεις σε θορυβώδεις περιοχές.
- *pattern* = 2 → Συμμετρική κατευθυντική ανίχνευση σε 3×3 πλέγμα
Εξετάζονται τέσσερις συμμετρικές κατευθύνσεις γύρω από κάθε pixel (οριζόντια, κάθετη, και οι δύο διαγώνιες). Για να θεωρηθεί μια θέση ως ακμή, απαιτείται τουλάχιστον 3 από τις 4 συμμετρικές κατευθύνσεις να παρουσιάζουν αλλαγή πρόσημου, και η διαφορά έντασης μεταξύ των αντίθετων σημείων να υπερβαίνει το κατώφλι *epsilon*. Αυτό το *pattern* μειώνει σημαντικά τα ψευδώς θετικά (εφαρμόζεται προαιρετικά **Gaussian smoothing** με $\sigma = 1.0$, πριν τον υπολογισμό του LoG, σύμφωνα με τη θεωρία του τελεστή).
- *pattern* = 3 → Αυστηρός έλεγχος σε 5×5 πλέγμα
Για κάθε pixel, ελέγχεται αν υπάρχει αντιστροφή πρόσημου μεταξύ ζευγών δομημένων γειτονικών περιοχών σε **τέσσερις κατευθύνσεις**: αριστερά–δεξιά, πάνω–κάτω, και στις δύο διαγώνιες. Ο έλεγχος περιλαμβάνει αξιολόγηση της συνέπειας πρόσημου σε πολλαπλά περιβάλλοντα σημεία (10+ ανά κατεύθυνση) και σύγκριση μέσων τιμών έντασης εκατέρωθεν της πιθανής ακμής. Το *pattern* = 3 είναι το πιο επιλεκτικό και κατάλληλο για περιπτώσεις που απαιτείται υψηλή ακρίβεια, όπως στην ανίχνευση γεωμετρικά ομαλών σχημάτων (π.χ. κύκλοι).

Η εικόνα εισόδου μετατρέπεται τελικά σε μια **δυναμική εικόνα εξόδου**, στην οποία κάθε pixel με τιμή 1 αντιστοιχεί σε σημείο ανίχνευσης ακμής, ενώ τα υπόλοιπα έχουν τιμή 0. Το κατώφλι *epsilon* χρησιμοποιείται για τον έλεγχο της διαφοράς μέσης έντασης εκατέρωθεν της πιθανής ακμής, ώστε να απορρίπτονται μη σημαντικές διελεύσεις. Μικρές τιμές δίνουν περισσότερες ακμές αλλά αυξάνουν τον θόρυβο. Υψηλότερες τιμές οδηγούν σε λιγότερες αλλά πιο αξιόπιστες ανιχνεύσεις.

Η συνάρτηση είναι πλήρως παραμετρική και μπορεί να εφαρμοστεί σε οποιαδήποτε δισδιάστατη εικόνα, με ελάχιστη απαίτηση την ύπαρξη επαρκούς τοπικής πληροφορίας για τον έλεγχο των γειτονιών

2.4 Αρχείο *circ_hough.py*

- *circ_hough*(*in_img_array*: *np.ndarray*, *R_max*: *float*, *dim*: *np.ndarray*, *V_min*: *int*) → *tuple*[*np.ndarray*, *np.ndarray*]:

Η συνάρτηση *circ_hough()* υλοποιεί τη **μέθοδο Hough** για την ανίχνευση κυκλικών περιγραμμάτων σε δυαδικές εικόνες ακμών. Η μέθοδος βασίζεται στην παραμετρική εξίσωση του κύκλου και στη συσσώρευση ψήφων (votes) σε έναν τρισδιάστατο χώρο παραμέτρων (κέντρο και ακτίνα).

Πιο συγκεκριμένα, για κάθε pixel της εικόνας εισόδου που έχει τιμή 1 (δηλαδή εντοπίστηκε ως ακμή), θεωρείται ότι μπορεί να ανήκει σε περιφέρεια κύκλων με διαφορετικά κέντρα και ακτίνες. Για κάθε ακτίνα εντός του διαστήματος $[R_{min}, R_{max}]$, δημιουργείται ένα πλήθος σημείων στον κύκλο, τα οποία αναπαριστούν πιθανά κέντρα κύκλων που περνούν από το συγκεκριμένο pixel. Οι συντεταγμένες αυτών των κέντρων προβάλλονται στον τρισδιάστατο πίνακα ψηφοφορίας (accumulator), αυξάνοντας την τιμή των αντίστοιχων κελιών.

Η διακριτοποίηση του παραμετρικού χώρου (σε ακτίνα και κέντρο) καθορίζεται από το διάνυσμα *dim*, το οποίο ορίζει πόσα σημεία εξετάζονται για κάθε διάσταση: κατά την κατακόρυφη (a), την οριζόντια (b) και την ακτίνα (r). Έτσι επιτυγχάνεται η διαχείριση υπολογιστικού κόστους και η προσαρμογή της μεθόδου σε διαφορετικά μεγέθη και ανάλυση εικόνων.

Μόλις ολοκληρωθεί η κατασκευή του πίνακα ψηφοφορίας, εφαρμόζεται μη μέγιστη καταστολή (non-maximum suppression) σε τοπικές γειτονιές, ώστε να αφαιρεθούν πολλαπλές ανιχνεύσεις για το ίδιο σχήμα και να διατηρηθούν μόνο τα τοπικά μέγιστα. Στη συνέχεια, εφαρμόζεται ένα κατώφλι V_{min} για τον αριθμό ψήφων, το οποίο χρησιμοποιείται για να φιλτραριστούν ψευδείς ανιχνεύσεις, διατηρώντας μόνο τα πιο αξιόπιστα μέγιστα (δηλ. σημεία με αρκετές ψήφους από την περιφέρεια των κύκλων).

Τελικά, οι υποψήφιες θέσεις (a, b, r) που πληρούν όλα τα παραπάνω κριτήρια μετατρέπονται σε πραγματικές συντεταγμένες στο επίπεδο της εικόνας και επιστρέφονται ως δύο πίνακες: ένας με τα κέντρα των κύκλων και ένας με τις αντίστοιχες ακτίνες. Το αποτέλεσμα είναι μία παραμετρικά εύχρηστη υλοποίηση του κυκλικού Hough Transform, ικανή να εντοπίζει κυκλικά σχήματα ανεξαρτήτως θέσης και μεγέθους, υπό την προϋπόθεση κατάλληλης ρύθμισης των υπερπαραμέτρων.

Η συνάρτηση είναι συμβατή με οποιαδήποτε δυαδική εικόνα ακμών, και μπορεί να συνεργαστεί με τελεστές ακμών όπως οι *sobel_edge()* και *log_edge()*

2.5 Αρχείο *demo.py*

Το αρχείο *demo.py* υλοποιεί την πρακτική εφαρμογή των συναρτήσεων που αναπτύχθηκαν στην εργασία, αξιοποιώντας την εικόνα *basketball_large.png* ως είσοδο. Ο στόχος του script είναι η οπτική αξιολόγηση των αποτελεσμάτων κάθε μεθόδου και η ανάλυση της επίδρασης των υπερπαραμέτρων τους.

Αρχικά, εφαρμόζεται η συνάρτηση *sobel_edge()* για πέντε διαφορετικές τιμές κατωφλίου (*thres*). Για κάθε περίπτωση (τιμή), αποθηκεύεται η αντίστοιχη εικόνα εξόδου (ακμών) και υπολογίζεται ο αριθμός των ανιχνευμένων pixels-ακμών, ώστε να διαπιστωθεί η επίδραση της παραμέτρου στην ευαισθησία ανίχνευσης. Επιπλέον, τα αποτελέσματα καταγράφονται γραφικά μέσω διαγράμματος που απεικονίζει τη μεταβολή του πλήθους ακμών ως συνάρτηση του κατωφλίου.

Στη συνέχεια, εφαρμόζεται η *log_edge()* στην ίδια εικόνα εισόδου μέσω της οποίας υπάρχει η δυνατότητα για αλλαγή στις παραμέτρους *pattern* και *epsilon* (βλ. 2.3), με σκοπό τη σύγκριση των αποτελεσμάτων με εκείνα του τελεστή Sobel. Παρατηρούνται διαφορές στην εντοπισμένη πληροφορία ακμών, λόγω της φύσης των δύο τελεστών.

Τέλος, εξετάζεται η ανίχνευση κύκλων μέσω της συνάρτησης *circ_hough()*, με είσοδο είτε από το αποτέλεσμα του *sobel_edge()* είτε από το αποτέλεσμα του *log_edge()* με *pattern = 3*. Η επιλογή της μεθόδου γίνεται μέσω μεταβλητών ελέγχου (*run_sobel, run_log*), επιτρέποντας την παραμετρική ενεργοποίηση μόνο ενός τελεστή κάθε φορά, ώστε να μειωθεί ο χρόνος εκτέλεσης και να διευκολυνθεί η αξιολόγηση.

- Για κάθε περίπτωση καθορίζεται διαφορετική τιμή παραμέτρων *dim* και *V_min*, προσαρμοσμένες στο είδος των ακμών που παράγει κάθε τελεστής. Η *dim* καθορίζει την ανάλυση του χώρου Hough ως προς τις παραμέτρους θέσης και ακτίνας, ενώ η *V_min* ορίζει το κατώφλι των ψήφων για να θεωρηθεί έγκυρος ένας κύκλος.
- Για κάθε τιμή *V_min*, το script εκτελεί την ανίχνευση και εμφανίζει τους ανιχνευμένους κύκλους πάνω στην αρχική εικόνα. Τα αποτελέσματα δείχνουν πώς μεταβάλλεται η ακρίβεια και πληρότητα της ανίχνευσης ανάλογα με τη μέθοδο ακμών και το κατώφλι ψήφων.

Διαπιστώνεται με χρήση της συνάρτησης *log_edge()* μόνο στην περίπτωση με *pattern = 3* εντοπίζονται ικανοποιητικά οι κύκλοι, καθώς ο αυστηρός έλεγχος μηδενικών διελεύσεων σε 5×5 γειτονιά οδηγεί σε καθαρές και καλά οριοθετημένες ακμές, απαραίτητες για επιτυχή χρήση του μετασχηματισμού Hough.

Το script παρέχει πλήρη κάλυψη των ζητούμενων, με παραμετρική χρήση και οπτικοποίηση των αποτελεσμάτων κάθε επιμέρους λειτουργίας.

3. Επίδειξη και σχολιασμός αποτελεσμάτων

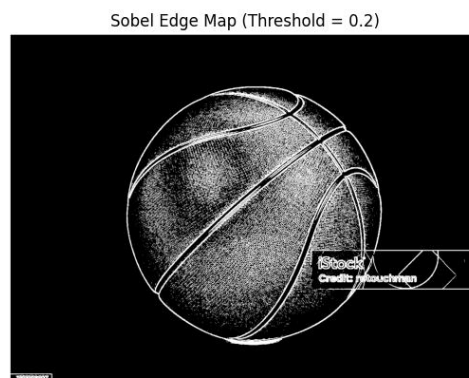
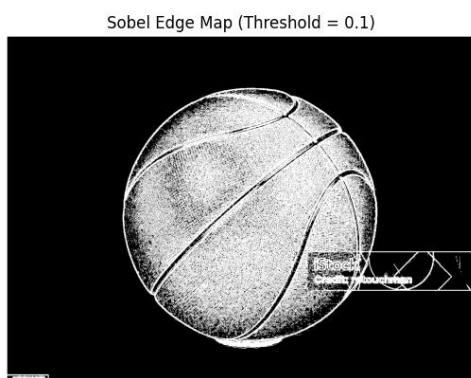
Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της εφαρμογής των υλοποιημένων συναρτήσεων στην εικόνα εισόδου *basketball_large.png* (παρουσιάζεται παρακάτω ως grayscale), με στόχο την αξιολόγηση της συμπεριφοράς κάθε μεθόδου και την κατανόηση της επίδρασης των παραμέτρων τους. Η μελέτη οργανώνεται σε τρία στάδια: ανίχνευση ακμών με **Sobel** και **LoG**, και εντοπισμός κύκλων με **Hough**.

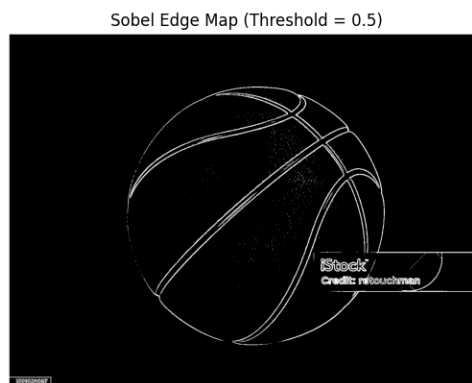
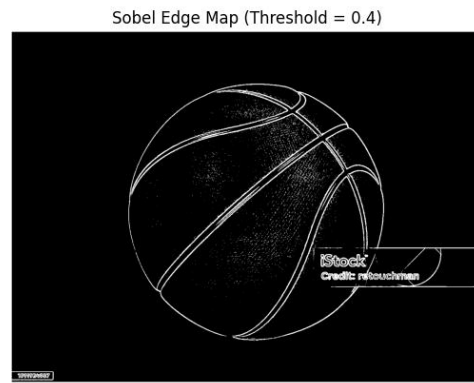
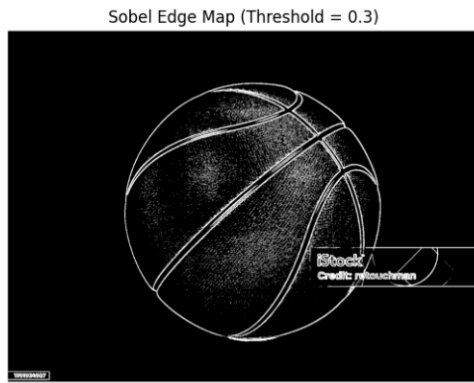


basketball_large.png (grayscale) 1

3.1 Ανίχνευση ακμών με τελεστή Sobel

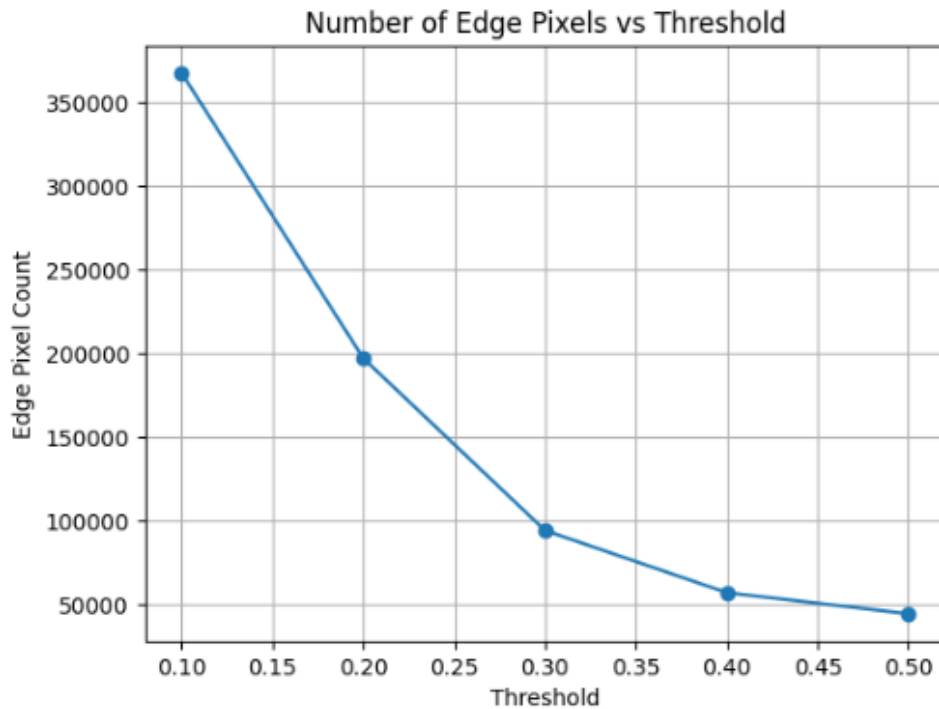
Στην παρούσα υποενότητα εξετάζεται η συμπεριφορά του τελεστή Sobel για την ανίχνευση ακμών, με έλεγχο της ευαισθησίας μέσω του κατωφλίου (threshold). Η συνάρτηση `sobel_edge` εφαρμόστηκε στην εικόνα εισόδου για πέντε διαφορετικές τιμές κατωφλίου: 0.1, 0.2, 0.3, 0.4 και 0.5, τα αποτελέσματα των οποίων φαίνονται παρακάτω:





- Όπως παρατηρείται από τις εικόνες, όσο μικρότερη είναι η τιμή του κατωφλίου, τόσο περισσότερα pixels χαρακτηρίζονται ως ακμές. Για παράδειγμα, με τιμή 0.1, ανιχνεύονται ακόμα και πολύ λεπτές διακυμάνσεις φωτεινότητας — αποτέλεσμα που οδηγεί σε υπερπλήρη (πυκνή) απεικόνιση, αλλά και σε αυξημένο θόρυβο και ψευδώς θετικά σημεία — ακόμα και εσωτερικές υφές της μπάλας αναδεικνύονται ως ακμές. Για τιμές 0.2 - 0.3 γίνεται πιο επιλεκτική ανίχνευση με τον θόρυβο να μειώνεται σταδιακά και τις εσωτερικές γραμμές της μπάλας να εμφανίζονται πιο καθαρά. Στις υψηλότερες τιμές 0.4 - 0.5, εντοπίζονται μόνο οι εντονότερες ακμές και υπάρχει πολύ καθαρή απεικόνιση των βασικών γεωμετρικών χαρακτηριστικών, όπως τα κύρια περιγράμματα της μπάλας και οι χαρακτηριστικές γραμμές της, αν και για 0.5 χάνεται μέρος της εσωτερικής πληροφορίας. Αυτό δείχνει ότι το threshold ρυθμίζει απευθείας τη «φιλτραρισμένη αυστηρότητα» του τελεστή Sobel και επιτρέπει έλεγχο ανάλογα με τις ανάγκες της εφαρμογής (π.χ. πληρότητα vs καθαρότητα ακμών).

Η επίδραση του κατωφλίου αποτυπώνεται και ποσοτικά στη γραφική παράσταση “*Number of Edge Pixels vs Threshold*”, η οποία παρουσιάζεται παρακάτω, όπου καταγράφεται η σχέση μεταξύ του πλήθους των ανιχνευμένων σημείων και της τιμής του κατωφλίου. Παρατηρείται έντονη φθίνουσα τάση, γεγονός που επιβεβαιώνει ότι το threshold αποτελεί κρίσιμη υπερπαραμέτρο ελέγχου της ευαισθησίας της μεθόδου.



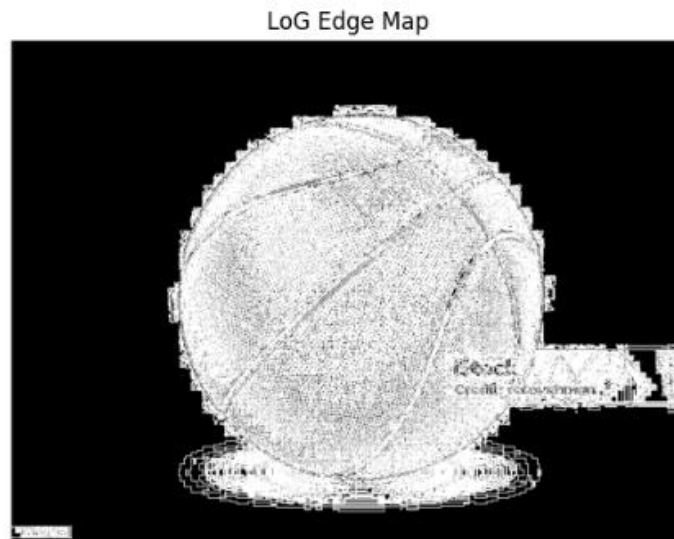
Για τις επόμενες ενότητες, επιλέγεται $\text{threshold} = 0.4$ ως συμβιβαστική επιλογή μεταξύ πληρότητας και ακρίβειας στην ανίχνευση των ακμών.

Παρά την πρακτική του αποτελεσματικότητα, ο τελεστής Sobel εμφανίζει περιορισμούς. Είναι ευαίσθητος σε θόρυβο, καθώς δεν ενσωματώνει μηχανισμούς φιλτραρίσματος, και η χρήση μικρών μάσκων 3×3 περιορίζει την ακρίβεια εντοπισμού των ακμών. Επίσης, η ανίχνευση είναι βέλτιστη μόνο σε οριζόντια και κάθετη κατεύθυνση, ενώ η απόδοση σε διαγώνιες ή καμπύλες ακμές είναι συγκριτικά μειωμένη.

3.2 Ανίχνευση ακμών με τελεστή LoG

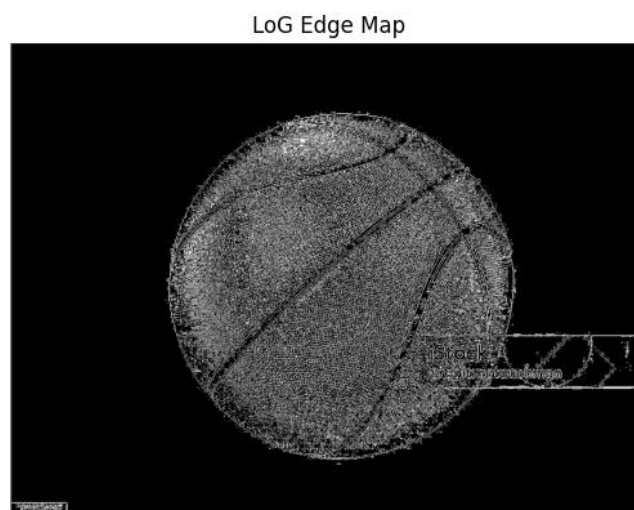
Η μέθοδος **LoG** εφαρμόστηκε για την ανίχνευση ακμών στο αποτέλεσμα της συνέλιξης της εικόνας εισόδου με διακριτή μάσκα Laplacian of Gaussian, με αξιολόγηση τριών διαφορετικών στρατηγικών εντοπισμού μηδενικών διελεύσεων (*zero-crossings*), όπως υλοποιήθηκαν στην `log_edge()` με βάση την τιμή της παραμέτρου *pattern*. Για κάθε περίπτωση παράχθηκε η αντίστοιχη δυαδική εικόνα ακμών, με τιμές 1 στα σημεία εντοπισμού και 0 αλλού.

Pattern 1: Απλή 3×3 ανίχνευση zero-crossing



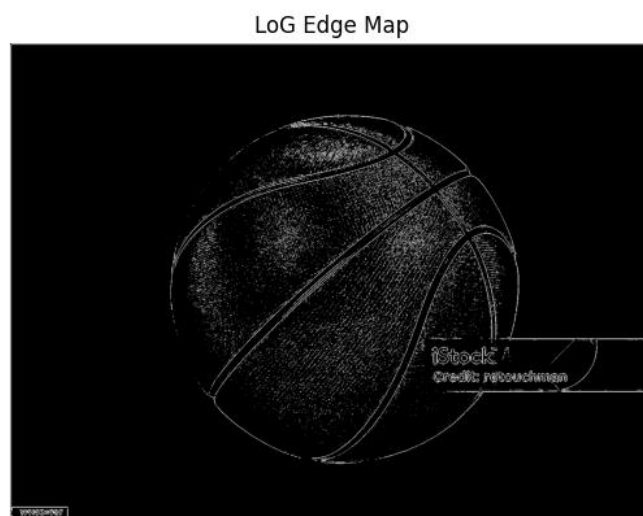
- Η πρώτη προσέγγιση εντοπίζει οποιαδήποτε αλλαγή πρόσημου στοπικό 3×3 παράθυρο. Το αποτέλεσμα εμφανίζει μεγάλο αριθμό αποσπασματικών ακμών και σημαντικό θόρυβο, ιδίως γύρω από το περίγραμμα της μπάλας και τις εσωτερικές της υφές. Η υπερευαισθησία της μεθόδου οδηγεί σε υπερανίχνευση, αλλά μειώνει τη χρησιμότητά της σε μεταγενέστερα στάδια, όπως το Hough.

Pattern 2: Συμμετρική κατευθυντική ανίχνευση (χωρίς smoothing)



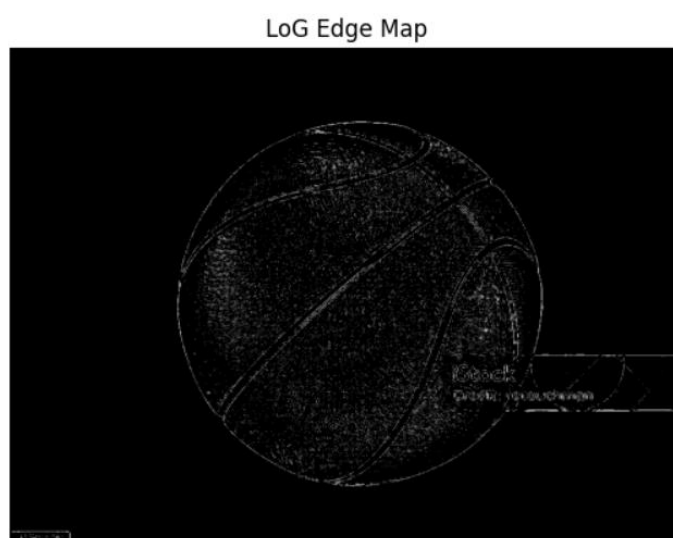
- Στην αρχική μορφή του pattern 2, δεν εφαρμόστηκε εξομάλυνση (smoothing) πριν τη συνέλιξη με τη μάσκα LoG. Το αποτέλεσμα είναι μεν ελαφρώς καθαρότερο από το pattern 1, αλλά εξακολουθεί να περιλαμβάνει θόρυβο, ιδίως εντός της επιφάνειας της μπάλας. Ο έλεγχος συμμετρικών διατάξεων βελτιώνει την απόρριψη ψευδών ακμών, αλλά δεν αρκεί χωρίς αποθορυβοποίηση.

Pattern 2: Συμμετρική κατευθυντική ανίχνευση (με Gaussian smoothing)



- Με την εφαρμογή Gaussian smoothing ($\sigma = 1.0$) πριν τον LoG, το αποτέλεσμα παρουσιάζει σαφώς βελτιωμένη καθαρότητα. Οι ακμές είναι ομαλές, συνεχείς και ακολουθούν με ακρίβεια τα γεωμετρικά όρια της μπάλας. Η υφή της επιφάνειας καταστέλλεται αποτελεσματικά, αποκαλύπτοντας τα σημαντικά περιγράμματα. Η μέθοδος αυτή είναι κατάλληλη για χρήση σε εφαρμογές όπως ανίχνευση σχημάτων με Hough.

Pattern 3: Αυστηρή ανίχνευση σε πλέγμα 5x5



- Η τρίτη στρατηγική εφαρμόζει αυστηρότερους ελέγχους zero-crossing με βάση τοπικές περιοχές 5x5 και μέσους όρους έντασης εκατέρωθεν. Το αποτέλεσμα είναι το πιο

«καθαρό» και γεωμετρικά σταθερό. Οι ακμές εμφανίζονται με καλή οριοθέτηση και ελάχιστο θόρυβο, αν και κάποιες πολύ λεπτές εσωτερικές λεπτομέρειες χάνονται. Αυτή η προσέγγιση ενδείκνυται για ακρίβεια σε περιγράμματα και για σενάρια όπου προέχει η αξιοπιστία έναντι της πληρότητας.

Σύγκριση της ανίχνευσης ακμών του τελεστή Sobel με τον τελεστή LoG

Η σύγκριση με τον τελεστή Sobel δείχνει ότι ο LoG, θεωρητικά, προσφέρει πιο ομοιόμορφη και γεωμετρικά συνεπή ανίχνευση περιγραμμάτων, καθώς και αυξημένη ανθεκτικότητα στον θόρυβο — ειδικά όταν εφαρμόζεται εξομάλυνση (Gaussian smoothing) ή χρησιμοποιούνται αυστηρότερα δομημένα κριτήρια (*pattern 2/3*). Ωστόσο, η αποτελεσματικότητα της μεθόδου εξαρτάται έντονα από την επιλογή παραμέτρων όπως το *epsilon* και η στρατηγική εντοπισμού μηδενικών διελεύσεων.

Στην πράξη, και για την εξεταζόμενη εικόνα, ο Sobel με $\text{threshold} = 0.4$ απέδωσε πιο καθαρές και συνεκτικές ακμές, ενώ ήταν σημαντικά απλούστερος στην παραμετροποίηση και ελαφρύτερος υπολογιστικά. Παρότι λιγότερο εξελιγμένος θεωρητικά, μπορεί να δώσει εξαιρετικά αποτελέσματα με κατάλληλη ρύθμιση.

Χαρακτηριστικό	Sobel	LoG
Ακρίβεια περιγραμμάτων	Πολύ καλή – συνεχείς, καθαρές γραμμές	Καλή - ιδίως με <i>pattern 2/3</i>
Αντοχή στον θόρυβο	Πολύ Καλή για $\text{thres} \geq 0.4$	Καλή με <i>smoothing</i> και αυστηρό <i>pattern</i>
Συνέχεια ακμών	Πολύ καλή σε βασικά περιγράμματα	Καλή – ομαλές και συνεκτικές ακμές
Καταλληλότητα για Hough	Καλή ήδη από $\text{thres} = 0.3 - 0.4$	Ικανοποιητική μόνο με <i>pattern 3</i>
Υπολογιστικό κόστος	Χαμηλό	Υψηλότερο (ειδικά με <i>grid-based</i> έλεγχο)
Παραμετροποίηση	Εύκολη (1 παράμετρος - <i>threshold</i>)	Πιο σύνθετη (συνδυασμός <i>pattern</i> + <i>epsilon</i>)

3.3 Ανίχνευση κύκλων με μετασχηματισμό Hough

Για την ανίχνευση κυκλικών σχημάτων στην εικόνα εισόδου χρησιμοποιήθηκε η συνάρτηση *circ_hough()*, η οποία υλοποιεί τον παραμετρικό μετασχηματισμό Hough για κύκλους. Η λειτουργία της βασίζεται στην παραμετρική εξίσωση του κύκλου και στη συσσώρευση ψήφων σε τρισδιάστατο πίνακα παραμέτρων (κέντρο και ακτίνα), όπως αναλύθηκε στη θεωρητική ενότητα 2.4.

Ως είσοδος στην *circ_hough()* χρησιμοποιήθηκαν δύο εναλλακτικοί χάρτες ακμών: το αποτέλεσμα της *sobel_edge()* με κατώφλι 0.4 και το αποτέλεσμα της *log_edge()* με χρήση της αυστηρής στρατηγικής *pattern = 3*. Στην πράξη, διαπιστώθηκε ότι μόνο η χρήση του LoG με *pattern = 3* οδήγησε σε επιτυχή και μοναδική ανίχνευση του κύκλου της μπάλας, ενώ οι υπόλοιπες μέθοδοι απέτυχαν.

Η ακρίβεια και πληρότητα της ανίχνευσης εξαρτήθηκε έντονα από τη ρύθμιση των υπερπαραμέτρων του μετασχηματισμού. Η παράμετρος *R_max* καθορίστηκε εμπειρικά σε τιμή 400, ώστε να καλύπτει πλήρως τον κύκλο της μπάλας, ενώ το *R_min* ορίστηκε στα 50 για να αγνοηθούν ανεπιθύμητοι μικροί κύκλοι θορύβου.

Η ανάλυση του παραμετρικού χώρου έγινε με τη ρύθμιση *dim = [200, 200, 125]*, το οποίο επιτρέπει ικανοποιητικό διαχωρισμό πιθανών συνδυασμών (κέντρο και ακτίνα), διατηρώντας ταυτόχρονα το κόστος υπολογισμού σε διαχειρίσιμα επίπεδα.

Στο σημείο αυτό να αναφέρω ότι παρουσιάζονται μόνο ενδεικτικά αποτελέσματα από κάθε κατηγορία εισόδου, καθώς οι περισσότερες παραλλαγές παρουσίασαν παρόμοια συμπεριφορά (ειδικά ως προς την ποιότητα ανίχνευσης). Διαφορές υπήρξαν κυρίως ως προς τη μέγιστη τιμή του accumulator και τον χρόνο εκτέλεσης.

Ακολουθεί σύγκριση των αποτελεσμάτων για διαφορετικές τιμές της παραμέτρου *V_min*, η οποία καθορίζει το ελάχιστο πλήθος ψήφων που απαιτούνται για να θεωρηθεί έγκυρη μία ανίχνευση κύκλου.

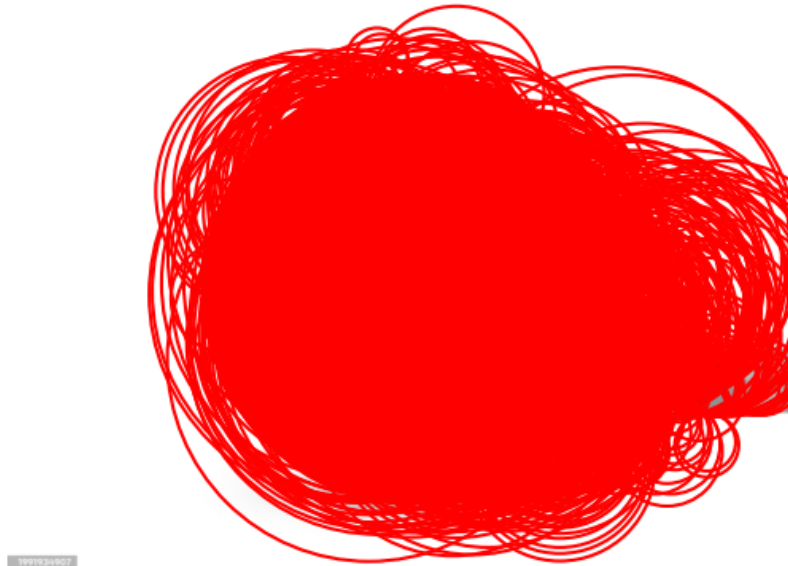
Πειραματικά αποτελέσματα και αξιολόγηση

❖ *log_edge()* με *pattern = 3* και *epsilon = 0.2*

Για να αξιολογηθεί η επίδοση του μετασχηματισμού Hough ως προς την παράμετρο ευαισθησίας *V_min*, εφαρμόστηκε η μέθοδος *circ_hough()* σε πέντε διαφορετικές τιμές: 200, 300, 400, 500 και 580. Οι τιμές της παραμέτρου *V_min* δεν επιλέχθηκαν αυθαίρετα, αλλά ρυθμίστηκαν αναλογικά με βάση το μέγιστο πλήθος ψήφων που καταγράφηκε στον πίνακα accumulator. Σε όλες τις περιπτώσεις χρησιμοποιήθηκε ως είσοδος το αποτέλεσμα

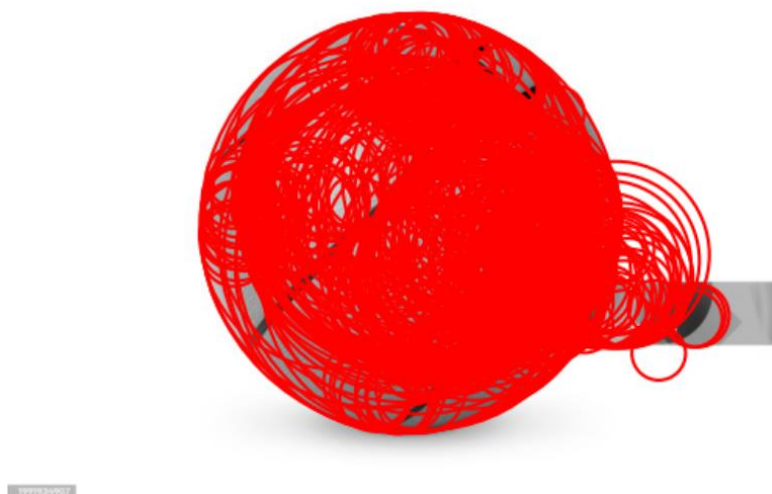
της `log_edge()` με `pattern = 3`, καθώς, όπως αναφέρθηκε, ήταν η μοναδική στρατηγική που παρήγαγε καθαρές, καλά οριοθετημένες και γεωμετρικά αξιόπιστες ακμές:

Hough Circles ($V_{\min} = 200$)



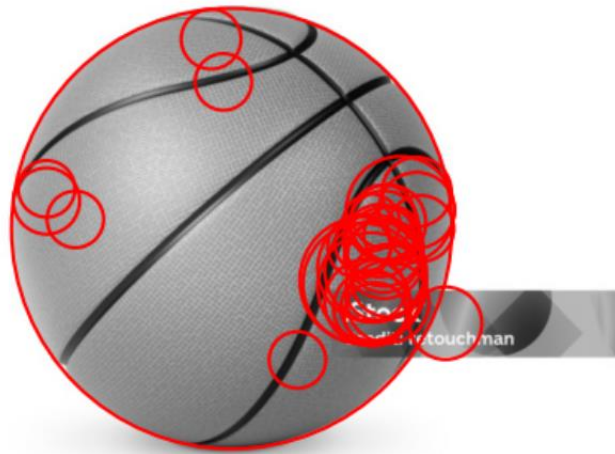
- Παρατηρείται πλήρης υπερανίχνευση. Ο μετασχηματισμός εντοπίζει εκατοντάδες ψευδείς κύκλους (false positives), με αποτέλεσμα ένα "σύννεφο" επικαλυπτόμενων ανιχνεύσεων χωρίς καμία αξιοπιστία. Αυτό οφείλεται στο χαμηλό όριο ψήφων που επιτρέπει την εμφάνιση κύκλων ακόμα και για τυχαία κατανομημένα σημεία ακμής.

Hough Circles ($V_{\min} = 300$)



- Η υπερανίχνευση παραμένει έντονη, αν και λιγότερη. Ο κύκλος της μπάλας δεν διακρίνεται ακόμα ξεκάθαρα, ενώ συνεχίζουν να εμφανίζονται πολλοί θόρυβοι στην περιφέρεια της εικόνας.

Hough Circles ($V_{\min} = 400$)



- Η μέθοδος εντοπίζει μερικούς αξιόπιστους κύκλους. Μειώνονται αισθητά τα ψευδώς ανιχνευμένα σχήματα και ο σωστός κύκλος αρχίζει να γίνεται ορατός, αλλά εξακολουθεί να πλαισιώνεται από πολλές μικρές ή λανθασμένες ανιχνεύσεις.

Hough Circles ($V_{\min} = 500$)



- Η ανίχνευση σταθεροποιείται. Ο κύκλος της μπάλας εντοπίζεται καθαρά και με ακρίβεια, ενώ οι περισσότερες ψευδείς ανιχνεύσεις έχουν εξαλειφθεί. Απομένει μία μικρή ανίχνευση κοντά στο δεξί άκρο, πιθανόν από υφή ή λάμψη.

Hough Circles ($V_{min} = 580$)



- Βέλτιστο αποτέλεσμα: ο μετασχηματισμός εντοπίζει μοναδικά τον σωστό κύκλο της μπάλας, με ιδανική προσαρμογή ως προς το κέντρο και την ακτίνα. Δεν υπάρχουν άλλες ανιχνεύσεις, αποδεικνύοντας ότι το συνδυαστικό φίλτρο LoG (*pattern 3*) + $V_{min} = 580$ αποτελεί τη βέλτιστη επιλογή για την παρούσα εικόνα.

Παρατηρήσεις:

- Σημαντική είναι και η παρακολούθηση της **μέγιστης τιμής** στον **πίνακα ψήφων**, η οποία στην περίπτωση της παρούσας εικόνας έφτασε τα 582. Αυτό υποδηλώνει ότι ο σωστός κύκλος συγκέντρωσε 582 ανεξάρτητες ψήφους από σημεία της περιφέρειάς του — δηλαδή σχεδόν όλα τα σημεία ακμής που εντοπίστηκαν μέσω του τελεστή LoG (*pattern 3*) συνέβαλαν στην ανίχνευσή του. Η πληροφορία αυτή αξιοποιήθηκε για την επιλογή της παραμέτρου V_{min} .
- Η συνολική **διάρκεια εκτέλεσης** της διαδικασίας ανίχνευσης κύκλων ήταν περίπου 30 λεπτά, λόγω του αυξημένου αριθμού υπολογισμών για τον εντοπισμό ακμών (δομημένος έλεγχος σε πλέγμα 5×5) και της υψηλής ανάλυσης του παραμετρικού χώρου ($\text{dim} = [200, 200, 125]$).

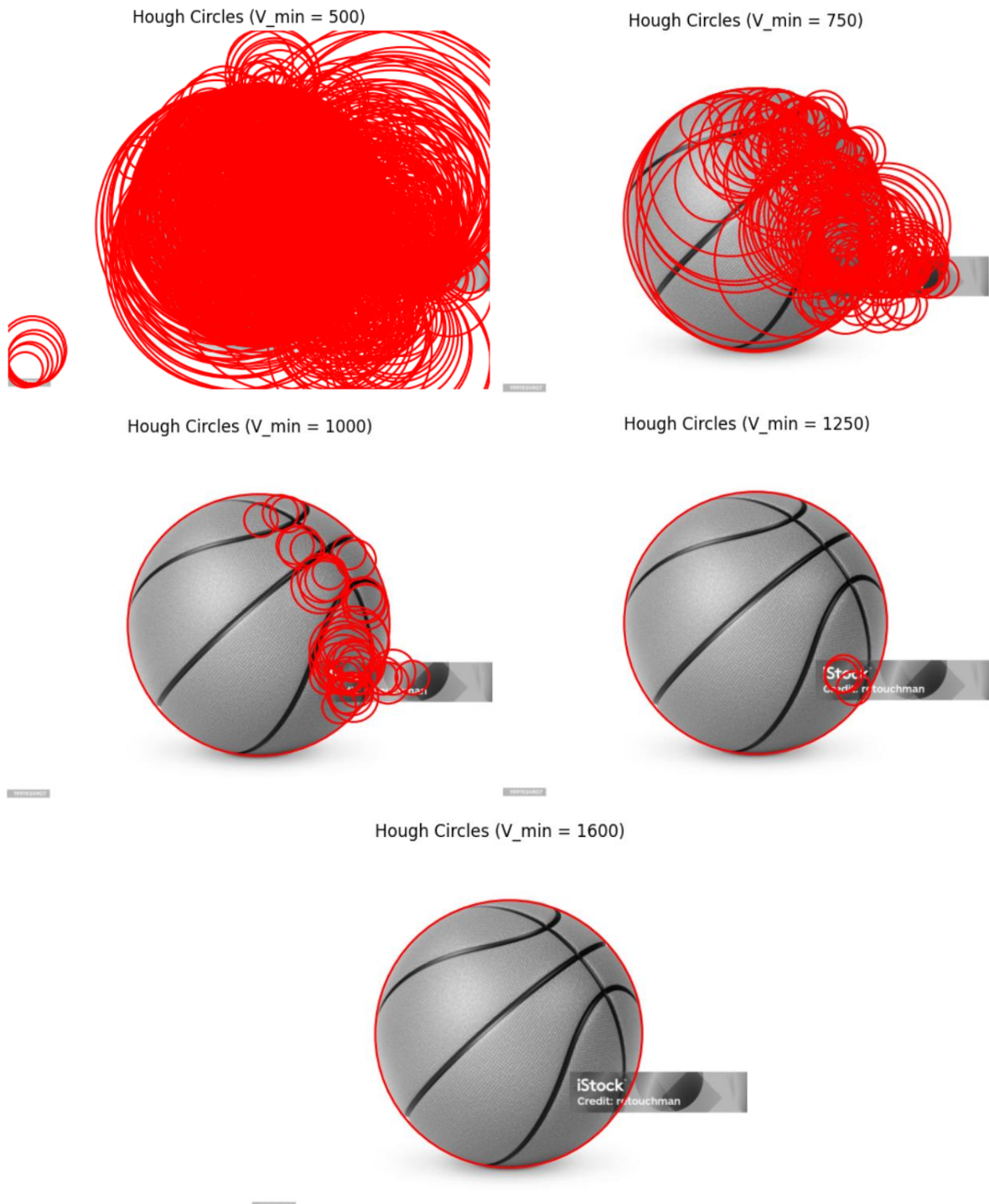
Συμπεράσματα:

- Η παράμετρος V_{min} είναι κρίσιμη για τον έλεγχο της ακρίβειας και της ευρωστίας του αλγορίθμου. Χαμηλές τιμές οδηγούν σε υπερανίχνευση, ενώ υψηλές τιμές περιορίζουν τις ανιχνεύσεις σε κύκλους υψηλής εμπιστοσύνης.
- Η μέθοδος αποδίδει πολύ καλά όταν οι ακμές της εικόνας είναι καθαρές και οι παράμετροι ρυθμιστούν σωστά.

❖ ***sobel_edge()* με *thres* = 0.4**

Για λόγους πληρότητας, πραγματοποιήθηκε και δοκιμή της μεθόδου *circ_hough()* με είσοδο το αποτέλεσμα της *sobel_edge()* (*threshold* = 0.4). Οι τιμές της παραμέτρου ορίστηκαν ως $V_{min} = [500, 750, 1000, 1250, 1600]$ και ρυθμίστηκαν αναλογικά με βάση το μέγιστο πλήθος ψήφων που καταγράφηκε στον πίνακα accumulator.

Τα αποτελέσματα ήταν παρόμοια με εκείνα της *log_edge()* από άποψη διάταξης ανιχνευμένων κύκλων, αλλά με σαφώς αυξημένο αριθμό επικαλύψεων. Για τον λόγο αυτό, παρουσιάζονται πιο συνοπτικά οι εικόνες:



Παρατηρήσεις:

- Ο **πίνακας ψήφων** (accumulator) εμφάνισε **μέγιστη τιμή 1641**, γεγονός που υποδηλώνει ότι υπήρχαν πολλά σημεία ακμών που στήριζαν κοινούς υποψήφιους κύκλους — όμως μεγάλο μέρος αυτών δεν αντιστοιχούσε στον πραγματικό κύκλο της μπάλας. Οι εικόνες εξόδου παρουσίασαν γενικά υπερανίχνευση, με πολλαπλές επικαλυπτόμενες ανιχνεύσεις και μειωμένη ακρίβεια.
- Παρά την απλούστερη φύση του τελεστή Sobel, ο **συνολικός χρόνος εκτέλεσης** της διαδικασίας ήταν περίπου 1 ώρα, δηλαδή σημαντικά μεγαλύτερος από την αντίστοιχη εκτέλεση με LoG (≈ 30 λεπτά). Αυτό οφείλεται στον **πολύ μεγαλύτερο αριθμό σημείων ακμής** που παρήγαγε ο Sobel, με αποτέλεσμα αυξημένο αριθμό ψήφων και επακόλουθο υπολογιστικό βάρος.

Γενική παρατήρηση για το "dim"

Η παραμετροποίηση του μετασχηματισμού Hough μέσω του διανύσματος $\text{dim} = [\text{num_a}, \text{num_b}, \text{num_r}]$ έχει άμεση και σημαντική επίδραση στον χρόνο εκτέλεσης και στην ακρίβεια εντοπισμού. Κατά τη διάρκεια των πειραμάτων δοκιμάστηκε και εναλλακτική ρύθμιση με $\text{dim} = [1442, 1070, 10]$ σε συνδυασμό με Sobel ως τελεστή ακμών. Σε αυτή την περίπτωση, ο μετασχηματισμός κάλυπτε πλήρως το εύρος των δυνατών κέντρων (μία τιμή ανά pixel της εικόνας), με μόλις 10 διακριτές ακτίνες. Η εκτέλεση ολοκληρώθηκε σε περίπου 9 λεπτά, γεγονός που δείχνει ότι η μείωση της τρίτης διάστασης (ακτίνας) μπορεί να επιταχύνει σημαντικά την εκτέλεση — με τίμημα, όμως, τη μείωση της ευκαμψίας στον εντοπισμό κύκλων διαφόρων μεγεθών.

Στη ρύθμιση αυτή, το μέγιστο πλήθος ψήφων στον πίνακα accumulator ήταν μόλις 67, και επιλέχθηκαν αντίστοιχα μικρές τιμές της παραμέτρου $V_{\text{min}} = [20, 30, 40, 50, 60]$. Η συνολική συμπεριφορά δείχνει ότι η ρύθμιση του dim είναι ένας κρίσιμος παράγοντας, που επηρεάζει τόσο το **υπολογιστικό κόστος** όσο και την **ικανότητα γενίκευσης** της μεθόδου σε κύκλους διαφόρων διαστάσεων.

4. Παραδοχές υλοποίησης

Κατά την ανάπτυξη των ζητούμενων συναρτήσεων και του script επίδειξης έγιναν ορισμένες τεχνικές παραδοχές και υλοποιήθηκαν επιλογές που στόχευαν είτε στη γενίκευση του κώδικα

είτε στη βελτίωση της ακρίβειας και σταθερότητας των αποτελεσμάτων, ενώ ανασταλτικό παράγοντα αποτέλεσαν και οι διαθέσιμοι υπολογιστικοί πόροι. Οι βασικές από αυτές είναι:

4.1 Γενική φιλοσοφία υλοποίησης

- Όλες οι συναρτήσεις (*sobel_edge()*, *log_edge()*, *circ_hough()*) υλοποιήθηκαν παραμετρικά, χωρίς χρήση εξωτερικών βιβλιοθηκών όπως OpenCV.
- Ο κώδικας είναι επαναχρησιμοποιήσιμος σε οποιαδήποτε grayscale εικόνα, εφόσον τηρούνται οι προϋποθέσεις της μεθόδου.

4.2 LoG και εντοπισμός ακμών

- Εφαρμόστηκε προαιρετική εξομάλυνση με Gaussian filter ($\sigma = 1.0$) πριν τη συνέλιξη με τη μάσκα LoG για ενίσχυση της σταθερότητας παρουσία θορύβου, αν και αυτό δεν απαιτούνταν ρητά στην εκφώνηση.
- Εξετάστηκαν τρία μοτίβα εντοπισμού μηδενικών διελεύσεων (*zero-crossings*):
 - Απλός έλεγχος πρόσημου σε 3×3 περιοχή
 - Συμμετρικά ζεύγη με έλεγχο έντασης ώστε να απορρίπτονται ασθενείς διελεύσεις. (*pattern 2*)
 - Πλήρης ανάλυση 5×5 με στατιστικό έλεγχο μέσω τιμών (*pattern 3*)
- Η παράμετρος *epsilon* (κατώφλι έντασης) δεν είναι δυναμική, αλλά ορίζεται προκαθορισμένα για κάθε δοκιμή (π.χ. *epsilon* = 0.2), με την μεταβολή της να έχει ελάχιστες αλλαγές στα αποτελέσματα.

4.3 παραδοχές στον Hough μετασχηματισμό

- Το εύρος ακτίνων ορίζεται σταθερά: $R_{min} = 50$, $R_{max} = 400$. Οι τιμές αυτές καθορίστηκαν εμπειρικά ώστε να αγνοούνται θόρυβοι μικρών κύκλων και να περιλαμβάνεται ο κύκλος της μπάλας (λόγω περιορισμένων υπολογιστικών πόρων).
- Το διάνυσμα *dim* ορίστηκε ως [200, 200, 125] για τις κύριες δοκιμές. Δοκιμάστηκαν και εναλλακτικές ρυθμίσεις, π.χ. [1442, 1070, 10], οι οποίες μείωσαν τον χρόνο εκτέλεσης αλλά περιόρισαν το εύρος ακτίνων.).
- Η παράμετρος V_{min} ρυθμίστηκε ανάλογα με το max value στον πίνακα accumulator (π.χ. $V_{min} = 580$ για LoG όπου max = 582, και $V_{min} = 60$ για Sobel με max = 67).
- Δεν εφαρμόστηκε περιορισμός περιοχής ενδιαφέροντος (margin filtering), ώστε να διατηρηθεί γενικότητα (υπάρχει μέσα στον κώδικα σε σχόλια και επιταχύνει αρκετά την εκτέλεση της μεθόδου).