COMPUTER VISION LAB EXERCISE 2

ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΑΡΑΪΣΚΟΣ
ΑΜ: 1072636

February 27, 2024

Ερώτημα 1

- imread: διαβάζει μία ασπρόμαυρη ή έγχρωμη εικόνα και επιστρέφει ένα μητρώο A διαστάσεων MxN για ασπρόμαυρες εικόνες ή ένα μητρώο MxNx3 για έγχρωμες εικόνες. Μπορούμε επίσης να διαβάσουμε μια συγκεκριμένη εικόνα(-ες) από ένα αρχείο που περιέχει πολλαπλές εικόνες (GIF, cursor file, etc). Για να διαβάσουμε εικόνες με επέκταση .png χρειάζεται να προσδιορίσουμε το 'BackgroundColor' ως 'none', ώστε οι διάφανες περιοχές να αναγνωριστούν ως διάφανες και να μην οριστεί κάποιο default Background Color.
- imwarp: εφαρμόζει έναν γεωμετρικό μετασχηματισμό σε μία εικόνα ή και μια απλή μετατόπιση. Κατά την εφαρμογή ενός μετασχηματισμού μπορούμε να εφαρμόσουμε και κάποιο είδος παρεμβολής, προκειμένου να υπολογίσουμε τις τιμές των pixel σε μη ακέραιες συντεταγμένες. Ακόμα, μπορούμε να ορίσουμε κάποιες επιπλέον παραμέτρους ώστε να ελέγχουμε διάφορα μέρη του γεωμετρικού μετασχηματισμού και της διαδικασίας στρέβλωσης της εικόνας.
- affine2d: δημιουργεί ένα αντικείμενο το οποίο αποθηκεύει πληροφορίες για έναν 2-διάστατο γεωμετρικό μετασχηματισμό, το οποίο έχει και τη δυνατότητα για προς τα εμπρός και ανάστροφους μετασχηματισμούς. Μετά την εφαρμογή του μετασχηματισμού οι παράλληλες ευθείες παραμένουν παράλληλες.
- projective2d: δημιουργεί ένα αντικείμενο το οποίο αποθηκεύει πληροφορίες για έναν 2-διάστατο προβολικό γεωμετρικό μετασχηματισμό. Μετά την εφαρμογή του μετασχηματισμού οι ευθείες γραμμές παραμένουν ευθείες.
- imref2d: δημιουργεί ένα 2-διάστατο αντικείμενο χωρικής αναφοράς για μία εικόνα. Δηλαδή δημιουργεί ένα χάρτη αντιστοίχισης μεταξύ των συντεταγμένων των pixel μιας εικόνας και τοποθεσιών του πραγματικού κόσμου.
- implay: αναπαράγει ταινίες, βίντεο ή και ακολουθίες εικόνων. Μπορούμε να αναπαράγουμε ολόκληρο το βίντεο, το πρώτο μόνο frame ενός βίντεο, καθώς και να επιλέξουμε την ταχύτητα εναλλαγής των frames.

Ερώτημα 2

Η εικόνα που θα δημιουργήσουμε, θα αποτελείται από κλιμακώσεις της εικόνας pudding.png. Η λογική που ακολουθήσαμε, αφού διαβάσουμε την εικόνα (pudding.png) και δημιουργήσουμε μια λευκή εικόνα για να τοποθετούμε πάνω σε αυτή τις κλιμακώσεις που θα παράγουμε είναι η παρακάτω. Παράγουμε έναν τυχαίο αριθμό, times, που καθορίζει τον αριθμό των κλιμακώσεων που θα τοποθετήσουμε στην εικόνα. Για times φορές, δημιουργούμε έναν affine2d μετασχηματισμό με τυχαία scaling στους άξονες x,y και τον εφαρμόζουμε με την imwarp στην εικόνα pudding. Παράγουμε 2 τυχαία m, n τα οποία αποτελούν το σημείο που θα τοποθετηθεί η νέα εικόνα πάνω στην εικόνα με όλες τις μετασχηματισμένες εικόνες. Δημιουργούμε τη μάσκα της scaled pudding εικόνας, επιλέγοντας τα σημεία με φωτεινότητα πάνω από 0.2, έτσι ώστε να κρατήσουμε μόνο την εικόνα και όχι το background. Τέλος, στα m, n της συνολικής εικόνας, προσθέτουμε την εικόνας της πουτίγκας, διατηρώντας την υπόλοιπη εικόνα.

Ερώτημα 3,4

Διαβάζουμε την εικόνα pudding.png, ορίζουμε το background της εικόνας να είναι λευκό και την μετατρέπουμε σε double μέσω της im2double, για καλύτερη διαχείρισή της. Έπειτα, δημιουργούμε ένα ημιτονοειδές σήμα, το οποίο θα δίνει τις συνεχείς τιμές που θέλουμε για προσομοιώσουμε τη στρέβλωση και ένα movie structure στο οποίο θα αποθηκεύουμε όλα τα frames που δημιουργούμε (ο αριθμός των frame είναι ίσος με τον αριθμό των τιμών που παίρνει η ημιτονοειδής συνάρτηση). Για αριθμό φορών ίσο με τον αριθμό των frames, δημιουργούμε μια εικόνα (που θα αποτελεί και το εκάστοτε frame). Η εικόνα αυτή έχει διπλάσιο πλάτος από την αρχική (pudding), καθώς όταν η στρέβλωση παίρνει ακραίες τιμές, το πλάτος της εικόνας που επιστρέφεται είναι διπλάσιο. Έπειτα, δημιουργούμε έναν affine2d μετασχηματισμό, του οποίου το μητρώο A παίρνει ως όρισμα στην κατεύθυνση x την τιμή του ημιτονοειδούς σήματος και στην κατεύθυνση y την τιμή 0, αφού θέλουμε η βάση του να παραμένει σταθερή. Τα scales για τους άξονες x, y παίρνουν τιμή 1 για να μείνουν αμετάβλητα. Εφαρμόζουμε το μετασχηματισμό στην εικόνα μέσω της imwarp και ορίζουμε να συμπληρώσει με 1 όσες τιμές απαιτούνται για να έχουμε λευκό background.
Για να τοποθετήσουμε την εικόνα χρειάζεται να προσαρμόσουμε το πλάτος της εικόνας (το μήκος δεν χρειάζεται κάποια αλλαγή). Όταν η στρέβλωση είναι θετική χρειάζεται να μετατοπίσουμε αριστερά την εικόνα, ενώ αν είναι αρνητική δεξιά και έπειτα τοποθετούμε στις συντεταγμένες m, n στην εικόνα που δημιουργήσαμε στην αρχή της for, την στρεβλωμένη εικόνα. Τέλος μετατρέπουμε την εικόνα σε frame, και το τοποθετούμε στο movie structure που δημιουργήσαμε.

Ερώτημα 5

Αρχικά διαβάζουμε τις εικόνες που θα χρειαστούμε (windmill, blades, blades_mask), ορίζουμε τον αριθμό των frames που θέλουμε να έχει το βίντεο που θα δημιουργήσουμε, δημιουργούμε ένα movie structure για να αποθηκεύσουμε το βίντεο και αρχικοποιούμε μια μεταβλητή η οποία θα συμβολίζει τον τρέχον αριθμό μοιρών που θα περιστραφούν οι λεπίδες του ανεμόμυλου και το βήμα με το οποίο θα αυξάνονται οι μοίρες.
Για να περιστρέφονται ως προς το κέντρο τους οι λεπίδες του ανεμόμυλου, θα πρέπει να μετατοπίσουμε το κέντρο των εικόνων (blades, blades_mask), από την πάνω αριστερή γωνία, στη μέση της. Αυτό το πετυχαίνουμε με την εφαρμογή ενός affine2d μετασχηματισμού, όπου οι τιμές tx και ty του μητρώου A παίρνουν τιμές m/2 και n/2 αντίστοιχα. Διατηρούμε σε μια μεταβλητή (imref_temp) το κέντρο της εικόνας, ώστε να το χρησιμοποιήσουμε ως σημείο αναφοράς στη συνέχεια.
Για κάθε frame, δημιουργούμε μια εικόνα η οποία περιέχει τον windmill και δημιουργούμε το μετασχηματισμό που θα πραγματοποιεί την περιστροφή. Η πρώτη γραμμή του μητρώου A αντιπροσωπεύει την περιστροφή ως προς τον άξονα x και η δεύτερη ως προς τον άξονα y. Δίνουμε ως όρισμα στην affine2d το ανάστροφο μητρώο A, γιατί η MATLAB χρησιμοποιεί row-major σειρά ενώ η συνάρτηση περιμένει column-major σειρά. Εφαρμόζουμε πάνω στη blades και blades_mask το μετασχηματισμό, δίνοντας και ως όρισμα το imref_temp, ώστε η περιστροφή να γίνεται ως προς το κέντρο της εικόνας.
Για να τοποθετήσουμε τις λεπίδες του ανεμόμυλου πάνω στην εικόνα, προβάλουμε την εικόνα με την imshow(), και με τη χρήση του εργαλείου data_tips επιλέξαμε ένα σημείο στην κορυφή του ανεμόμυλου. Με βάση το σημείο του ανεμόμυλου που επιλέξαμε και των διαστάσεων των λεπίδων, ορίσουμε τα m, n, δηλαδή το σημείο όπου θα τοποθετηθεί η πάνω αριστερή γωνία της εικόνας των λεπίδων. Για τα m, n στην εικόνα του ανεμόμυλου, εφαρμόζουμε τη μάσκα, ώστε να κρατήσουμε όλη την εικόνα, εκτός των σημείων που θα προστεθούν οι λεπίδες και σε αυτά τα σημεία που αφαιρέθηκαν προσθέτουμε την εικόνα με τις λεπίδες. Τέλος, αυξάνουμε τη γωνία κατά το βήμα που ορίσαμε και αποθηκεύουμε την εικόνα στο movie structure που ορίσαμε, αφού πρώτα τη μετατρέψουμε σε frame με την im2frame.

Ερώτημα 6

Με τη χρήση της imwarp() μπορούμε να χρησιμοποιήσουμε 3 μεθόδους παρεμβολής, linear, cubic και nearest. Όποια μέθοδο και να χρησιμοποιήσουμε δεν γίνεται αισθητή με το μάτι η διαφορά στην ποιότητα του βίντεο γιατί οι εικόνες που χρησιμοποιούμε καθώς και το τελικό βίντεο έχει αρκετά χαμηλή ανάλυση, με αποτέλεσμα να αποδίδουν όλες ισοδύναμα.
Η ιδανική μεταξύ αυτών των 3, είναι η cubic γιατί δίνει καλύτερα αποτελέσματα αλλά είναι και η πιο ακριβή σε υπολογιστικό κόστος. Η linear είναι η default μέθοδος που χρησιμοποιείται καθώς δίνει καλά αποτελέσματα χωρίς να έχει μεγάλο υπολογιστικό κόστος και η nearest (-neighbor) η οποία είναι η απλούστερη καθώς συμπληρώνει τις τιμές των pixel με αυτή του κοντινότερού του και έχει επίσης πολύ χαμηλό υπολογιστικό κόστος.

Ερώτημα 7

Αρχικά διαβάζουμε τις εικόνες που θα χρειαστούμε για τη δημιουργία του βίντεο (beach.jpg, ball.jpg, ball_mask.jpg). Για να μπορούμε να μετατοπίζουμε τη μπάλα ως προς το κέντρο της, εφαρμόζουμε έναν affine2d μετασχηματισμό με τιμές tx=m/2 και ty=n/2, όπου m,n οι διαστάσεις της εικόνας της μπάλας και διατηρούμε σε μία μεταβλητή (ball_ref και ball_mask_ref) το κέντρο των εικόνων. Καθώς η εικόνα της μπάλας είναι αρκετά μεγάλη, την μικραίνουμε κατά ¼ ώστε να χωρέσει στην εικόνα της παραλίας.
Για να αναπαραστήσουμε την κίνηση της μπάλας να αναπηδά πάνω στην παραλία θα δημιουργήσουμε μία φθίνουσα συνάρτηση συνημιτόνου με βάση την οποία θα μετατοπίζουμε την μπάλα πάνω στην εικόνα της παραλίας. Ακόμα δημιουργούμε και ένα struct τα οποίο θα περιέχει τα frames του βίντεο.
Για τη δημιουργία του βίντεο δημιουργούμε μία for loop μέσα στην οποία ακολουθείται η εξής διαδικασία: περιστρέφουμε την μπάλα μερικές μοίρες, υπολογίζουμε την μετατόπιση της μπάλας ως προς την παραλία, υπολογίζουμε τις συντεταγμένες της παραλίας που θα τοποθετήσουμε την μπάλα, δημιουργούμε το frame (αφαιρούμε τη μάσκα της μπάλας από την παραλία και προσθέτουμε στις ίδιες συντεταγμένες την μπάλα) και τέλος ανανεώνουμε το struct που δημιουργήσαμε για να αποθηκεύσουμε το βίντεο.

Ερώτημα 8

Για να δημιουργήσουμε την αίσθηση ότι η μπάλα "χάνεται" προς τον ορίζοντα όσο αναπηδά, ακολουθήσαμε την ίδια διαδικασία με το προηγούμενο ερώτημα και προσθέσαμε 2 παραμέτρους, το scale_factor και το ball_step. Με την πρώτη μεταβλητή μειώνουμε το μέγεθος της μπάλας κατά ένα σταθερό παράγοντα και με τη δεύτερη μειώνουμε την συνάρτηση συνημιτόνου που δημιουργήσαμε. Με αυτό τον τρόπο, το σημείο μηδενισμού της συνάρτησης, δηλαδή εκεί που θα χτυπούσε η μπάλα στο έδαφος, μετατοπίζεται συνεχώς λίγο πιο πάνω, ώστε να φαίνεται λες και μπάλα μετατοπίζεται στο βάθος της εικόνας. Η λογική πίσω από την υλοποίηση των υπολοίπων παραμένει η ίδια με το προηγούμενο ερώτημα.