

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

**ΘΕΜΑΤΑ ΟΡΑΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΓΡΑΦΙΚΗΣ**

**ΑΝΑΦΟΡΑ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ**

**3Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**

ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ ΠΟΤΑΜΙΑΝΟΣ

Α.Μ. 1084537

up1084537@ac.upatras.gr

Πάτρα, 2024

ΑΣΚΗΣΗ 1.

1. **Ανίχνευση Ακροτάτων στον Χώρο Κλίμακας**Εδώ, δημιουργείται η πυραμίδα των εικόνων (Gaussian και DoG) και υπολογίζονται οι διαφορές των Γκαουσιανών.  
   Πιο συγκεκριμένα, από την αρχή του παραπάνω σχολίου μέχρι το σημείο πριν την επεξεργασία των ακροτάτων, ο κώδικας δημιουργεί τα Gaussian blurred images και τα DoG images. Αυτό περιλαμβάνει τις γραμμές:

%% Scale-Space Extrema Detection  
tic  
% original sigma and the number of actave can be modified. the larger  
% sigma0, the more quickly-smooth images  
sigma0=sqrt(2);  
octave=3;%6\*sigma\*k^(octave\*level)<=min(m,n)/(2^(octave-2))  
level=3;  
D=cell(1,octave);  
**for** i=1:octave  
D(i)=mat2cell(zeros(row\*2^(2-i)+2,colum\*2^(2-i)+2,level),row\*2^(2-i)+2,colum\*2^(2-i)+2,level);  
**end**  
% first image in first octave is created by interpolating the original one.  
temp\_img=kron(img,ones(2));  
temp\_img=padarray(temp\_img,[1,1],'replicate');  
figure(2)  
subplot(1,2,1);  
imshow(origin)  
%create the DoG pyramid.  
**for** i=1:octave  
…  
 **end**  
 D{i}=temp\_D;  
 temp\_img=temp\_img(1:2:**end**,1:2:**end**);  
 temp\_img=padarray(temp\_img,[1,1],'both','replicate');  
**end**  
toc

1. **Εύρεση Τοπικών Σημείων-Κλειδιών**

Μετά τον υπολογισμό των DoG, ο κώδικας εντοπίζει τα τοπικά ακρότατα στον χώρο κλίμακας. Σε αυτό το τμήμα γίνεται η σύγκριση κάθε pixel με τους 26 γείτονές του (9 στην προηγούμενη κλίμακα, 9 στην τρέχουσα και 9 στην επόμενη κλίμακα) για να εντοπιστούν τα τοπικά ακρότατα. Στη συνέχεια εφαρμόζονται έλεγχοι για την απόρριψη σημείων που δεν έχουν επαρκή αντίθεση ή βρίσκονται σε άκρα εικόνων.

%% Keypoint Localisation  
% search each pixel in the DoG map to find the extreme point  
tic  
interval=level-1;  
number=0;  
**for** i=2:octave+1  
 number=number+(2^(i-octave)\*colum)\*(2\*row)\*interval;  
**end**  
extrema=zeros(1,4\*number);  
flag=1;  
**for** i=1:octave  
…  
…  
…  
%% accurate keypoint localization   
%eliminate the point with low contrast or poorly localised on an edge  
% x:|,y:-- x is for vertial and y is for horizontal  
% value comes from the paper.  
tic  
threshold=0.1;  
r=10;  
extr\_volume=length(extrema)/4;  
[m,n]=size(img);  
secondorder\_x=conv2([-1,1;-1,1],[-1,1;-1,1]);  
secondorder\_y=conv2([-1,-1;1,1],[-1,-1;1,1]);  
**for** i=1:octave  
 **for** j=1:level  
 test=D{i}(:,:,j);  
 temp=-1./conv2(test,secondorder\_y,'same').\*conv2(test,[-1,-1;1,1],'same');  
 D{i}(:,:,j)=temp.\*conv2(test',[-1,-1;1,1],'same')\*0.5+test;  
 **end**  
**end**  
**for** i=1:extr\_volume  
…  
…  
…  
subplot(2,2,4);  
imshow(origin)  
hold on  
plot(ry,rx,'b+');  
toc

Εδώ γίνεται η λεπτομερής εύρεση ακριβών θέσεων των σημείων κλειδιών και απόρριψη των σημείων που δεν πληρούν τα κριτήρια αντίθεσης και καμπυλότητας της επιφάνειας.  
  
**3. Διόρθωση Κατεύθυνσης**  
Μετά τον εντοπισμό των σημείων κλειδιών, στο επόμενο στάδιο αποδίδεται προσανατολισμός σε κάθε σημείο-κλειδί, ώστε ο περιγραφέας που θα υπολογιστεί αργότερα να είναι αμετάβλητος ως προς την περιστροφή.

%% Orientation Assignment(Multiple orientations assignment)  
tic  
kpori=zeros(1,36\*extr\_volume);  
minor=zeros(1,36\*extr\_volume);  
f=1;  
flag=1;  
**for** i=1:extr\_volume  
…  
…  
…  
 **end**  
**end**  
idx= minor==0;  
minor(idx)=[];  
extrema=minor;  
% delete unsearchable points and add minor orientation points  
idx= kpori==0;  
kpori(idx)=[];  
extr\_volume=length(extrema)/4;  
toc

1. **Περιγραφέας Σημείων-Κλειδιών**

Αφού πλέον κάθε σημείο κλειδί έχει θέση, κλίμακα και προσανατολισμό, ακολουθεί ο υπολογισμός του περιγραφέα. Ο περιγραφέας SIFT βασίζεται στην κατασκευή προσανατολισμένων ιστογραμμάτων κλίσης σε υποπεριοχές γύρω από το σημείο-κλειδί. Εδώ υπολογίζεται το 128-διάστατο διάνυσμα που περιγράφει τοπικά το σημείο κλειδί. Με αυτό τον τρόπο τα σημεία μπορούν να αντιστοιχιστούν μεταξύ διαφορετικών εικόνων.

%% keypoint descriptor  
tic  
d=4;% In David G. Lowe experiment,divide the area into 4\*4.  
pixel=4;  
feature=zeros(d\*d\*8,extr\_volume);  
**for** i=1:extr\_volume  
 descriptor=zeros(1,d\*d\*8);% feature dimension is 128=4\*4\*8;  
 width=d\*pixel;  
 %x,y centeral point and prepare for location rotation  
 x=floor((extrema(4\*(i-1)+3)-1)/(n/(2^(extrema(4\*(i-1)+1)-2))))+1;  
 y=mod((extrema(4\*(i-1)+3)-1),m/(2^(extrema(4\*(i-1)+1)-2)))+1;  
 z=extrema(4\*(i-1)+2);  
 **if**((m/2^(extrema(4\*(i-1)+1)-2)-pixel\*d\*sqrt(2)/2)>x&&x>(pixel\*d/2\*sqrt(2))&&(n/2^(extrema(4\*(i-1)+1)-2)-pixel\*d/2\*sqrt(2))>y&&y>(pixel\*d/2\*sqrt(2)))  
 sub\_x=(x-d\*pixel/2+1):(x+d\*pixel/2);  
 sub\_y=(y-d\*pixel/2+1):(y+d\*pixel/2);  
 sub=zeros(2,length(sub\_x)\*length(sub\_y));  
 j=1;  
 **for** p=1:length(sub\_x)  
 **for** q=1:length(sub\_y)  
 sub(:,j)=[sub\_x(p)-x;sub\_y(q)-y];  
 j=j+1;  
 **end**  
 **end**  
  
…  
…  
…  
index=find(sum(feature));  
feature=feature(:,index);  
toc

**ΑΣΚΗΣΗ 2.**

**1. Ανίχνευση Ακροτάτων στον Χώρο Κλίμακας (Scale-Space Extrema Detection)**

**Βήματα:**

1. **Δημιουργία του Χώρου Κλίμακας:**
   * Για την αρχική εικόνα I(x, y), δημιουργούμε πολλαπλές "οκτάβες" (octaves).
   * Κάθε οκτάβα περιλαμβάνει σειρά εικόνων λείων με διαφορετικές τιμές σ (διάφορα επίπεδα κλίμακας).
   * Οι λείες εικόνες προκύπτουν με συνέλιξη της αρχικής εικόνας με Γκαουσιανά φίλτρα διαφορετικής διασποράς σ.
2. **Υπολογισμός Διαφοράς Γκαουσιανών (DoG):**
   * Για κάθε οκτάβα, από τις λείες εικόνες αφαιρείται η μία από την επόμενη, ώστε να προκύψουν οι εικόνες Difference-of-Gaussian (DoG).
   * Κάθε DoG εικόνα δημιουργείται χρησιμοποιώντας ένα σταθερό παράγοντα κλίμακας k.
3. **Προετοιμασία για Εντοπισμό Τοπικών Ακροτάτων:**
   * Οι DoG εικόνες οργανώνονται σε πυραμίδα με οκτάβες και επίπεδα.
   * Αυτό θα χρησιμοποιηθεί στο επόμενο στάδιο για ανίχνευση ακροτάτων.

**2. Εύρεση Τοπικών Σημείων-Κλειδιών (Keypoint Localization)**

**Βήματα:**

1. **Εντοπισμός Τοπικών Ακροτάτων:**
   * Κάθε εικονοστοιχείο σε κάθε DoG εικόνα συγκρίνεται με τους 8 γείτονές του στην ίδια κλίμακα και τους 18 γείτονες στις γειτονικές κλίμακες (συνολικά 26 γείτονες).
   * Ένα εικονοστοιχείο θεωρείται ακρότατο (τοπικό μέγιστο ή ελάχιστο) αν η τιμή του είναι είτε μεγαλύτερη είτε μικρότερη από όλες αυτές των 26 γειτόνων.
2. **Ακριβής Προσδιορισμός Θέσης Σημείου Κλειδιού:**
   * Χρησιμοποιείται μία προσέγγιση με επέκταση Taylor γύρω από την αρχική θέση ακροτάτου, ώστε να προσδιοριστεί η ακριβής θέση του στο συνεχές.
   * Με αυτόν τον τρόπο βρίσκουμε πιο ακριβή θέση σε υποεικονοστοιχειακό επίπεδο.
3. **Απόρριψη Ακατάλληλων Σημείων:**
   * Απορρίπτονται σημεία χαμηλής αντίθεσης (threshold στο μέγεθος του DoG).
   * Απορρίπτονται σημεία που βρίσκονται πάνω σε άκρα (edges), χρησιμοποιώντας τον λόγο των ιδιοτιμών του Hessian (Edge response test).

**3. Διόρθωση Κατεύθυνσης (Orientation Assignment)**

**Βήματα:**

1. **Υπολογισμός Τοπικών Κλίσεων γύρω από το Σημείο Κλειδί:**
   * Υπολογίζεται η κλίση μεγέθους m(x, y) και κατεύθυνσης θ(x, y) σε μια γειτονιά γύρω από το σημείο κλειδί, στην κατάλληλη κλίμακα.
2. **Κατασκευή Ιστογράμματος Κατευθύνσεων:**
   * Τα διανύσματα κλίσης τακτοποιούνται σε ένα ιστογράφημα 36 bins που καλύπτουν όλες τις πιθανές γωνίες (0° έως 360°).
   * Κάθε μέγεθος κλίσης σταθμίζεται και από έναν Γκαουσιανό πυρήνα ανάλογα με την απόσταση από το σημείο κλειδί.
3. **Εύρεση Κύριας και Συμπληρωματικών Κατευθύνσεων:**
   * Η κύρια κατεύθυνση ορίζεται ως η γωνία που αντιστοιχεί στο μέγιστο του ιστογράμματος.
   * Όποιες άλλες κορυφές στο ιστογράφημα που ξεπερνούν το 80% της κύριας κορυφής οδηγούν στη δημιουργία επιπλέον σημείων κλειδιού με τις αντίστοιχες κατευθύνσεις.

**4. Υπολογισμός του Τοπικού Περιγραφέα Σημείου Κλειδιού (Keypoint Descriptor)**

**Βήματα:**

1. **Κανονικοποίηση Έναντι Κλίμακας και Προσανατολισμού:**
   * Η περιοχή γύρω από το σημείο κλειδί περιστρέφεται σύμφωνα με την κύρια κατεύθυνση ώστε ο περιγραφέας να είναι αμετάβλητος ως προς την περιστροφή.
   * Λαμβάνεται η κατάλληλη κλίμακα γύρω από το σημείο ώστε να είναι αμετάβλητος και ως προς την αλλαγή κλίμακας.
2. **Δημιουργία Υποπεριοχών και Υπολογισμός Ιστογραμμάτων Κλίσης:**
   * Χωρίζεται η περιοχή γύρω από το σημείο κλειδί σε ένα πλέγμα 4x4 υποπεριοχών.
   * Για κάθε υποπεριοχή, υπολογίζεται ένα ιστογράφημα 8 προσανατολισμών.
   * Συνολικά προκύπτουν 4x4x8 = 128 τιμές, που αποτελούν το χαρακτηριστικό διάνυσμα του περιγραφέα.
3. **Κανονικοποίηση του Περιγραφέα:**
   * Το 128-διάστατο διάνυσμα κανονικοποιείται ώστε να είναι ανεξάρτητο από την παγκόσμια φωτεινότητα ή την αντίθεση.
   * Αποκόπτονται μεγάλες τιμές (clipping), ακολουθεί εκ νέου κανονικοποίηση για επιπλέον σταθερότητα.

**ΑΣΚΗΣΗ 3**  
Σύμφωνα με τον κώδικα SIFT\_feature.m που δόθηκε και την τυπική υλοποίηση του αλγορίθμου SIFT, ο αριθμός των επιπέδων (levels) ανά οκτάβα είναι συνήθως 3. Στο συγκεκριμένο παράδειγμα κώδικα που εξετάσαμε, χρησιμοποιείται η εντολή:

**level** = 3;

η οποία ορίζει ότι ανά οκτάβα παράγονται 3 επίπεδα κλίμακας για τον υπολογισμό των διαφορών Gaussians (DoG). Έτσι, η απάντηση είναι ότι χρησιμοποιούνται 3 επίπεδα ανά οκτάβα.  
  
**ΑΣΚΗΣΗ 4**  
  
**1. Παρεμβολή με Πολυωνική Προσέγγιση (Polynomial Fitting)**

Μετά τον εντοπισμό ενός σημείου κλειδιού σε συγκεκριμένη κλίμακα, μπορούμε να εφαρμόσουμε μία πολυωνική προσαρμογή (π.χ. τριών διαστάσεων για x, y, κλίμακα σ) των τιμών της DoG. Με αυτόν τον τρόπο, προσεγγίζουμε τη συνάρτηση απόκρισης γύρω από το σημείο και επιλύοντας τις μερικές παραγώγους (διαφορές) βρίσκουμε τη θέση του μέγιστου ή ελάχιστου με υποεικονοστοιχειακή (sub-pixel) και υποκλίμακα (sub-scale) ακρίβεια.

**Πλεονεκτήματα:**

* Αυξημένη ακρίβεια στον προσδιορισμό της θέσης και της κλίμακας.
* Μειώνει την επίδραση του θορύβου, καθώς η πολυωνική προσαρμογή λειαίνει τις τιμές.

**Μειονεκτήματα:**

* Αυξημένο υπολογιστικό κόστος σε σύγκριση με την απλή διακριτοποίηση.
* Ευαισθησία στην επιλογή του πολυωνικού βαθμού: πολύ χαμηλός βαθμός δεν προσφέρει σημαντική βελτίωση, πολύ υψηλός μπορεί να οδηγήσει σε υπερπροσαρμογή (overfitting).

**2. Χρήση Μεθόδου Επέκτασης Taylor (Taylor Expansion)**  
Στην υλοποίηση του SIFT χρησιμοποιείται ήδη μία τρισδιάστατη επέκταση Taylor γύρω από το αρχικά εντοπισμένο ακρότατο. Μπορούμε να εφαρμόσουμε μια πιο ακριβή ή επαναληπτική διαδικασία προσέγγισης Taylor, επεκτείνοντας τους υπολογισμούς ή εκτελώντας περισσότερα βήματα βελτιστοποίησης.

**Πλεονεκτήματα:**

* Εφαρμόζεται σχετικά εύκολα καθώς ήδη χρησιμοποιείται στο βασικό SIFT.
* Δίνει κλειστή μορφή λύσεων για τη μετατόπιση κλίμακας και θέσης.

**Μειονεκτήματα:**

* Εξαρτάται από την ορθότητα του μοντέλου Taylor· αν η συνάρτηση DoG δεν είναι καλά προσεγγίσιμη από ένα πολυώνυμο χαμηλού βαθμού, η ακρίβεια περιορίζεται.
* Προσθέτει επιπλέον επαναλήψεις και υπολογιστικό κόστος.

**3. Χρήση Μεθόδων Βελτιστοποίησης (Optimization Techniques)**  
Εφαρμογή αριθμητικών μεθόδων βελτιστοποίησης, όπως ο αλγόριθμος Newton-Raphson ή Levenberg-Marquardt, πάνω στην συνάρτηση DoG στον χώρο κλίμακας. Ξεκινούμε από την αρχική εκτίμηση και βελτιώνουμε επαναληπτικά τη θέση και την κλίμακα έως ότου συγκλίνουμε σε ένα τοπικό μέγιστο/ελάχιστο.

**Πλεονεκτήματα:**

* Μεγάλη ευελιξία: μπορούμε να συνδυάσουμε διαφορετικές συναρτήσεις κόστους, τακτικοποιήσεις ή περιορισμούς.
* Δυνατότητα εύρεσης πιο ακριβούς λύσης εφόσον το πρόβλημα είναι καλά ορισμένο.

**Μειονεκτήματα:**

* Αυξημένο υπολογιστικό κόστος.
* Πιθανότητα παγίδευσης σε τοπικά ακρότατα εάν η αρχική εκτίμηση ή το μοντέλο δεν είναι επαρκώς καλές.
* Απαιτεί καλή ρύθμιση παραμέτρων (π.χ. μήκους βήματος, κριτήρια σύγκλισης).

**4. Αυξημένη Πυκνότητα Δειγματοληψίας στην Κλίμακα (Over-Sampling in Scale Space)**  
Αντί να χρησιμοποιείται ένας συγκεκριμένος παράγοντας k για την αλλαγή κλίμακας μεταξύ των επιπέδων, μπορούμε να παράγουμε περισσότερες ενδιάμεσες εικόνες (π.χ. 4 ή 5 επίπεδα ανά οκτάβα αντί για 3) ώστε να έχουμε πιο πυκνή δειγματοληψία και άρα μικρότερο κενό μεταξύ διαδοχικών κλιμάκων.

**Πλεονεκτήματα:**

* Απλή μέθοδος που δεν απαιτεί περίπλοκη μαθηματική μοντελοποίηση.
* Επιτυγχάνει καλύτερη κατά προσέγγιση ακρίβεια χωρίς πολύπλοκη βελτιστοποίηση.

**Μειονεκτήματα:**

* Αυξάνει σημαντικά τον υπολογισμό (περισσότερες εικόνες, περισσότερος χρόνος επεξεργασίας).
* Δεν δίνει πραγματικά υποκλίμακα ακρίβεια, απλώς πυκνώνει το "πλέγμα" στον άξονα κλίμακας.

**5. Προσέγγιση με Πολυδιάστατη Ενδοδιάμεση Παρεμβολή (Multi-Dimensional Interpolation)**  
Χρήση ενδοδιάμεσης παρεμβολής (interpolation) με ανώτερες spline συναρτήσεις στον τρισδιάστατο χώρο (x, y, κλίμακα). Με αυτό τον τρόπο μπορούμε να εκτιμήσουμε τις τιμές DoG σε μη ακέραια επίπεδα κλίμακας και να βρούμε ακριβέστερα το ακρότατο.

**Πλεονεκτήματα:**

* Πολύ ομαλή προσέγγιση της συνάρτησης.
* Δυνατότητα πολύ υψηλής ακρίβειας.

**Μειονεκτήματα:**

* Πολύπλοκη υλοποίηση και ακόμη υψηλότερο υπολογιστικό κόστος.
* Ευαισθησία σε εξωγενείς παραμορφώσεις (π.χ. θόρυβος, κακή αρχικοποίηση).

**ΑΣΚΗΣΗ 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| liftingbody | moon | tire | rice | westconcordorthophoto |
|  |  |  |  |  |

Επιλέγω ασπρόμαυρες φωτογραφίες λόγω περιορισμού από το directory: \Matlab\toolbox\images\imdata :  
liftingbody.png,moon.tif, tire.tif, rice.png, westconcordorthophoto.png

1. **liftingbody.png:**

A close-up of a white object

Description automatically generatedA white object in the air

Description automatically generated

Στις εικόνες βλέπουμε τα αποτελέσματα της εφαρμογής του αλγορίθμου SIFT.  
Συγκεκριμένα:

**Εικόνα 1 (πράσινα σημεία):** Παρατηρούμε την αρχική ανίχνευση των σημείων-κλειδιών (keypoints) πριν την εφαρμογή των φίλτρων βελτιστοποίησης. Τα σημεία συγκεντρώνονται κυρίως στις περιοχές υψηλής αντίθεσης και στα όρια του αεροσκάφους, όπου υπάρχουν έντονες μεταβολές στην κλίμακα του γκρι. Αυτό είναι αναμενόμενο καθώς ο αλγόριθμος SIFT εντοπίζει χαρακτηριστικά σημεία βασιζόμενος στις διαφορές Γκαουσιανών (DoG).

**Εικόνα 2 (μπλε σημεία):** Μετά την εφαρμογή των φίλτρων βελτιστοποίησης και την απόρριψη των ασταθών σημείων, παρατηρούμε μια πιο επιλεκτική κατανομή των σημείων-κλειδιών. Τα εναπομείναντα σημεία αντιπροσωπεύουν πιο σταθερά χαρακτηριστικά της εικόνας, όπως γωνίες και ακμές με υψηλή αντίθεση. Αυτά τα σημεία είναι πιο αξιόπιστα για εφαρμογές όπως η αναγνώριση αντικειμένων ή η σύγκριση εικόνων.

1. **moon.tif:**

A moon with green dots

Description automatically generatedA close up of the moon

Description automatically generated

Στην πρώτη εικόνα (πράσινα σημεία), παρατηρούμε την αρχική ανίχνευση των σημείων-κλειδιών από τον αλγόριθμο. Είναι εμφανές ότι τα περισσότερα σημεία εντοπίζονται κατά μήκος του φωτεινού περιγράμματος της Σελήνης, όπου υπάρχει η μέγιστη αντίθεση μεταξύ του σκοτεινού φόντου και της φωτεινής επιφάνειας. Αυτό είναι απολύτως αναμενόμενο, καθώς ο αλγόριθμος SIFT βασίζεται στον εντοπισμό περιοχών με έντονες διαφοροποιήσεις στην ένταση των εικονοστοιχείων μέσω των διαφορών Γκαουσιανών (DoG).

Στη δεύτερη εικόνα (με μπλε σημεία), βλέπουμε το αποτέλεσμα μετά την εφαρμογή των φίλτρων βελτιστοποίησης. Παρατηρούμε ότι διατηρούνται κυρίως τα σημεία στο έντονο περίγραμμα της Σελήνης, ενώ έχουν αφαιρεθεί αρκετά σημεία από το εσωτερικό της επιφάνειας που πιθανώς οφείλονταν σε θόρυβο ή σε περιοχές χαμηλής αντίθεσης. Αυτό καταδεικνύει την αποτελεσματικότητα του αλγορίθμου στην επιλογή των πιο σταθερών και αξιόπιστων χαρακτηριστικών σημείων.

Ιδιαίτερο ενδιαφέρον παρουσιάζει ο τρόπος με τον οποίο ο αλγόριθμος εντόπισε επίσης μερικά σημεία-κλειδιά στις περιοχές των κρατήρων και των σκιάσεων στην επιφάνεια της Σελήνης, αναδεικνύοντας την ικανότητά του να αναγνωρίζει τοπικά χαρακτηριστικά πέρα από το κύριο περίγραμμα του αντικειμένου.

1. **tire.tif:**

A close-up of a tire

Description automatically generated

Στην πρώτη εικόνα (πράσινα σημεία), παρατηρούμε την αρχική ανίχνευση των σημείων-κλειδιών. Ο αλγόριθμος έχει εντοπίσει πληθώρα σημείων που ακολουθούν τη χαρακτηριστική κυκλική γεωμετρία του τροχού. Είναι αξιοσημείωτο πώς τα σημεία συγκεντρώνονται κατά μήκος των ομόκεντρων κύκλων της ζάντας, καθώς και στα σημεία όπου υπάρχουν έντονες εναλλαγές φωτεινότητας μεταξύ των μεταλλικών στοιχείων και των σκοτεινών διαστημάτων.

Στη δεύτερη εικόνα (μπλε σημεία), βλέπουμε το αποτέλεσμα μετά την εφαρμογή των φίλτρων βελτιστοποίησης. Παρατηρούμε ότι έχει διατηρηθεί η κυκλική κατανομή των σημείων-κλειδιών, αλλά με μεγαλύτερη επιλεκτικότητα. Τα εναπομείναντα σημεία αντιπροσωπεύουν πιο σταθερά χαρακτηριστικά, ιδιαίτερα στις περιοχές όπου υπάρχουν διακριτά γεωμετρικά μοτίβα της ζάντας.

Συγκρίνοντας με την αρχική, μπορούμε να εκτιμήσουμε καλύτερα πώς ο αλγόριθμος SIFT έχει εντοπίσει αποτελεσματικά τα βασικά δομικά στοιχεία του τροχού. Ιδιαίτερα σημαντική είναι η ανίχνευση σημείων τόσο στην εξωτερική περιφέρεια του ελαστικού όσο και στα εσωτερικά στοιχεία της ζάντας, δημιουργώντας ένα ολοκληρωμένο σύνολο χαρακτηριστικών

1. **rice.png**

A green and white pattern

Description automatically generated with medium confidenceA close-up of a grey background

Description automatically generated

Κατά την αρχική φάση ανίχνευσης (απεικόνιση με πράσινα σημεία), ο αλγόριθμος εντόπισε σημεία-κλειδιά που αντιστοιχούν στις περιοχές υψηλής βαθμίδας έντασης, κυρίως στα όρια μεταξύ των κόκκων και του υποβάθρου. Η χωρική κατανομή των σημείων αυτών ακολουθεί την ανισότροπη γεωμετρία των μεμονωμένων κόκκων, αναδεικνύοντας τη μορφολογική τους ομοιογένεια.

Μετά την εφαρμογή των κριτηρίων βελτιστοποίησης (απεικόνιση με μπλε σημεία), παρατηρείται σημαντική μείωση του πλήθους των σημείων-κλειδιών, με διατήρηση εκείνων που παρουσιάζουν υψηλή διακριτική ικανότητα ως προς τα τοπικά χαρακτηριστικά της εικόνας. Η επιλεκτική αυτή διατήρηση των σημείων υποδηλώνει την αποτελεσματικότητα του αλγορίθμου στην αναγνώριση σταθερών χαρακτηριστικών ανεξάρτητων από μετασχηματισμούς κλίμακας και περιστροφής.

1. **westconcordorthophoto.png**

Green lines on a tree

Description automatically generated with medium confidenceA blue cross on a white background

Description automatically generated with medium confidence

Στην αρχική φάση (πράσινα σημεία), ο αλγόριθμος εντόπισε πληθώρα σημείων-κλειδιών που αντιστοιχούν κυρίως στις διασταυρώσεις οδών, τις γωνίες κτιρίων και τα όρια διαφορετικών χρήσεων γης. Η πυκνότητα των σημείων είναι ιδιαίτερα υψηλή στις περιοχές με έντονη αστική δραστηριότητα, όπου παρατηρούνται απότομες μεταβολές στην ένταση των εικονοστοιχείων.

Μετά την εφαρμογή των κριτηρίων βελτιστοποίησης (μπλε σημεία), παρατηρείται μια πιο επιλεκτική κατανομή των σημείων-κλειδιών. Ο αλγόριθμος διατήρησε κυρίως τα σημεία που αντιστοιχούν σε σταθερά γεωμετρικά χαρακτηριστικά της αστικής δομής, όπως οι κύριες διασταυρώσεις και τα διακριτά όρια μεταξύ διαφορετικών τύπων κάλυψης γης.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η συγκέντρωση σημείων-κλειδιών κατά μήκος των κύριων οδικών αρτηριών, υποδεικνύοντας την ικανότητα του αλγορίθμου να αναγνωρίζει γραμμικά στοιχεία υψηλής σημασίας στον αστικό ιστό.

**ΑΣΚΗΣΗ 6**

Η συγκεκριμένη άσκηση με δυσκόλεψε με αποτέλεσμα να χρειαστεί να πάρω παραδοχή. Δεν την υλοποίησα κάνοντας χρήση της SIFT\_feature.m αλλά με χρήση της συνάρτησης detectSIFTFeatures για την οποία κατάφερα να βρω αντίστοιχο documentation. Συνημμένα ωστόσο θα βρείτε και τα match\_SIFT\_RANSAC.m, SIFT\_feature\_func.m τα οποία κάνουν απόπειρα να υλοποιήσουν ακριβώς την άσκηση, αδυνατώντας ωστόσο να τρέξει για διαφορετικές εικόνες (δηλαδή κάνει αντιστοίχιση σημείων σε 2 όμοιες εικόνες).

**Ανίχνευση Σημείων-Κλειδιών SIFT**

Χρησιμοποιούνται οι συναρτήσεις detectSIFTFeatures και extractFeatures για την ανίχνευση και εξαγωγή των χαρακτηριστικών SIFT από τις δύο εικόνες:

points1 = detectSIFTFeatures(img1\_gray);  
points2 = detectSIFTFeatures(img2\_gray);  
  
[features1, validPoints1] = extractFeatures(img1\_gray, points1);  
[features2, validPoints2] = extractFeatures(img2\_gray, points2);

Αυτό επιτρέπει την εξαγωγή μοναδικών περιγραφέων για κάθε σημείο-κλειδί, που είναι απαραίτητα για την αντιστοίχιση.

**Αντιστοίχιση Περιγραφέων**

Η αντιστοίχιση των περιγραφέων πραγματοποιείται με χρήση της συνάρτησης matchFeatures, εφαρμόζοντας το Ratio Test για την αποφυγή ψευδών αντιστοιχιών:

indexPairs = matchFeatures(**features1**, features2, 'MaxRatio', 0.8, 'Unique', true);  
  
matchedPoints1 = validPoints1(**indexPairs**(:,1));  
matchedPoints2 = validPoints2(**indexPairs**(:,2));

Ο Ratio Test περιορίζει τις αντιστοιχίσεις στο 80% της αναλογίας μεταξύ του πρώτου και δεύτερου κοντινότερου γείτονα, αυξάνοντας την αξιοπιστία των αντιστοιχιών.

**Εκτίμηση Γεωμετρικής Μετασχηματισμού με RANSAC**

Για την απομάκρυνση ακατάλληλων αντιστοιχιών (outliers), εφαρμόζεται ο αλγόριθμος RANSAC για την εκτίμηση της γεωμετρικής μετασχηματισμού μεταξύ των δύο συνόλων σημείων:

[tform, inlierIdx] = estimateGeometricTransform2D(matchedPoints1, matchedPoints2, ...  
 'projective', 'MaxDistance', 3);  
  
inlierPoints1 = matchedPoints1(inlierIdx, :);  
inlierPoints2 = matchedPoints2(inlierIdx, :);

Ο RANSAC επιτρέπει την ανίχνευση των inliers που συμφωνούν με το υποτιθέμενο μοντέλο μετασχηματισμού, βελτιώνοντας την ακρίβεια της αντιστοίχισης.

**match\_SIFT\_RANSAC\_buildin.m:**

% match\_SIFT\_RANSAC\_buildin.m  
%   
% Αυτό το σκριπτ υλοποιεί την αντιστοίχιση σημείων-κλειδιών μεταξύ δύο εικόνων  
% χρησιμοποιώντας τον αλγόριθμο SIFT και RANSAC για την απομάκρυνση ακατάλληλων αντιστοιχίσεων.  
%  
% Προαπαιτούμενα:  
% - Computer Vision Toolbox  
%  
% Χρήση:  
% - Αντικαταστήστε τα 'image1.jpg' και 'image2.jpg' με τα ονόματα των εικόνων σας.  
%  
% Συγγραφέας: ΑΓΓΕΛΟΣ ΝΙΚΟΛΑΟΣ ΠΟΤΑΜΙΑΝΟΣ  
  
%% Αρχικοποίηση  
clear; clc; close all;  
  
%% Ορισμός Ονομάτων Εικόνων  
image1\_filename = 'cameraman.tif'; % Αντικαταστήστε με το όνομα της πρώτης εικόνας  
image2\_filename = 'cameraman\_zoom.png'; % Αντικαταστήστε με το όνομα της δεύτερης εικόνας  
  
%% Φόρτωση και Προεπεξεργασία Εικόνων  
fprintf('Φόρτωση και προεπεξεργασία εικόνων...\n');  
  
% Φόρτωση πρώτης εικόνας  
img1 = imread(image1\_filename);  
**if** size(img1,3) == 3  
 img1\_gray = rgb2gray(img1);  
**else**  
 img1\_gray = img1;  
**end**  
img1\_gray = imresize(img1\_gray, [256, 256]);  
img1\_gray = im2double(img1\_gray);  
  
% Φόρτωση δεύτερης εικόνας  
img2 = imread(image2\_filename);  
**if** size(img2,3) == 3  
 img2\_gray = rgb2gray(img2);  
**else**  
 img2\_gray = img2;  
**end**  
img2\_gray = imresize(img2\_gray, [256, 256]);  
img2\_gray = im2double(img2\_gray);  
  
%% Ανίχνευση Σημείων-Κλειδιών SIFT  
fprintf('Ανίχνευση σημείων-κλειδιών SIFT...\n');  
points1 = detectSIFTFeatures(img1\_gray);  
points2 = detectSIFTFeatures(img2\_gray);  
  
% Εξαγωγή Περιγραφέων  
[features1, validPoints1] = extractFeatures(img1\_gray, points1);  
[features2, validPoints2] = extractFeatures(img2\_gray, points2);  
  
fprintf('Αριθμός σημείων-κλειδιών στην πρώτη εικόνα: %d\n', size(features1,1));  
fprintf('Αριθμός σημείων-κλειδιών στη δεύτερη εικόνα: %d\n', size(features2,1));  
  
%% Αντιστοίχιση Περιγραφέων  
fprintf('Αντιστοίχιση περιγραφέων με χρήση Ratio Test...\n');  
indexPairs = matchFeatures(features1, features2, 'MaxRatio', 0.8, 'Unique', true);  
  
matchedPoints1 = validPoints1(indexPairs(:,1));  
matchedPoints2 = validPoints2(indexPairs(:,2));  
  
fprintf('Αριθμός αντιστοιχιών μετά το Ratio Test: %d\n', size(matchedPoints1,1));  
  
%% Εκτίμηση Geometric Transform με RANSAC  
fprintf('Εφαρμογή του αλγορίθμου RANSAC για εύρεση inliers...\n');  
[tform, inlierIdx] = estimateGeometricTransform2D(matchedPoints1, matchedPoints2, ...  
 'projective', 'MaxDistance', 3);  
  
inlierPoints1 = matchedPoints1(inlierIdx, :);  
inlierPoints2 = matchedPoints2(inlierIdx, :);  
  
fprintf('Αριθμός inliers μετά το RANSAC: %d\n', size(inlierPoints1,1));  
  
%% Οπτικοποίηση Τελικών Αντιστοιχιών  
fprintf('Οπτικοποίηση των τελικών αντιστοιχιών...\n');  
  
% Δημιουργία συνενωμένης εικόνας (πλευρική σύνδεση) με ασπρόμαυρες εικόνες  
img1\_display = imresize(img1\_gray, [256, 256]);  
img2\_display = imresize(img2\_gray, [256, 256]);  
  
% Διασφάλιση ότι και οι δύο εικόνες έχουν το ίδιο πλήθος καναλιών  
**if** size(img1\_display, 3) == 1  
 img1\_display = cat(3, img1\_display, img1\_display, img1\_display);  
**end**  
  
**if** size(img2\_display, 3) == 1  
 img2\_display = cat(3, img2\_display, img2\_display, img2\_display);  
**end**  
  
% Συνένωση των εικόνων  
combined\_img = [img1\_display, img2\_display];  
  
% Προσαρμογή των συντεταγμένων της δεύτερης εικόνας  
inlierPoints2\_shifted = inlierPoints2;  
inlierPoints2\_shifted.Location(:,1) = inlierPoints2\_shifted.Location(:,1) + size(img1\_display,2);  
  
% Δημιουργία νέας μορφής για εμφάνιση  
figure;  
imshow(combined\_img);  
hold on;  
  
% Σχεδίαση inlier σημείων  
plot(inlierPoints1.Location(:,1), inlierPoints1.Location(:,2), 'ro', 'MarkerSize',5, 'LineWidth',1.5);  
plot(inlierPoints2\_shifted.Location(:,1), inlierPoints2\_shifted.Location(:,2), 'go', 'MarkerSize',5, 'LineWidth',1.5);  
  
% Σχεδίαση γραμμών μεταξύ των inliers  
**for** i = 1:size(inlierPoints1,1)  
 line([inlierPoints1.Location(i,1), inlierPoints2\_shifted.Location(i,1)], ...  
 [inlierPoints1.Location(i,2), inlierPoints2\_shifted.Location(i,2)], 'Color', 'y');  
**end**  
  
title('Αντιστοιχίσεις Σημείων-Κλειδιών με Inliers από RANSAC');  
hold off;  
  
fprintf('Η αντιστοίχιση ολοκληρώθηκε. Ελέγξτε το παράθυρο της εικόνας για τα αποτελέσματα.\n');

A person using a camera

Description automatically generated

Εικόνα 1: cameraman.tif και cameraman\_zoom.png

A person with a camera

Description automatically generated

Εικόνα 2: cameraman.tif και cameraman\_rotate.png

A person shooting a picture

Description automatically generated

Εικόνα 3: cameraman\_zoom.png και cameraman\_rotate.png

**ΑΣΚΗΣΗ 7**

**Βασικές Αρχές του RANSAC:**

Ο αλγόριθμος ξεκινά με την τυχαία δειγματοληψία ενός υποσυνόλου δεδομένων που είναι απαραίτητο για την εκτίμηση του μοντέλου. Ο αριθμός των σημείων που επιλέγονται εξαρτάται από τον τύπο του μοντέλου που θέλουμε να εκτιμήσουμε (π.χ., δύο σημεία για ευθεία, τρία για επίπεδο). Στη συνέχεια, χρησιμοποιείται αυτό το υποσύνολο για να εκτιμηθούν οι παράμετροι του μοντέλου. Μετά την εκτίμηση, ο αλγόριθμος ελέγχει όλα τα δεδομένα για να προσδιορίσει ποια σημεία συμφωνούν με το εκτιμημένο μοντέλο εντός ενός προκαθορισμένου κατωφλίου. Το μοντέλο που έχει τον μεγαλύτερο αριθμό "inliers" (σημεία που συμφωνούν με το μοντέλο) θεωρείται το καλύτερο. Αυτά τα βήματα επαναλαμβάνονται για έναν προκαθορισμένο αριθμό επαναλήψεων ή μέχρι να βρεθεί ένα μοντέλο που πληροί τα κριτήρια αποδοχής.

**(+) Πλεονεκτήματα του RANSAC:**

Ο RANSAC είναι ιδιαίτερα ανθεκτικός στα outliers, καθώς μπορεί να λειτουργήσει αποτελεσματικά ακόμα και όταν ένα μεγάλο ποσοστό των δεδομένων αποτελεί outliers. Επιπλέον, είναι ευέλικτος και μπορεί να εφαρμοστεί σε διάφορα μοντέλα και προβλήματα, καθιστώντας τον κατάλληλο για πολλές εφαρμογές.

**(-) Μειονεκτήματα του RANSAC:**

Ωστόσο, ο αλγόριθμος έχει και ορισμένα μειονεκτήματα. Η τυχαία δειγματοληψία μπορεί να απαιτεί μεγάλο αριθμό επαναλήψεων, ειδικά σε περιπτώσεις υψηλού ποσοστού outliers, γεγονός που αυξάνει τον υπολογιστικό χρόνο. Επιπλέον, η απόδοση του RANSAC εξαρτάται από την επιλογή των παραμέτρων, όπως το κατώφλι αποδοχής και ο αριθμός των επαναλήψεων, οι οποίοι μπορεί να μην είναι πάντα εύκολο να καθοριστούν βέλτιστα.

**Χρήση του RANSAC στον Αλγόριθμο Αντιστοίχισης SIFT**

Στον αλγόριθμο match\_SIFT\_RANSAC\_buildin.m, ο RANSAC χρησιμοποιείται για την εκτίμηση της γεωμετρικής μετασχηματιστικής σχέσης μεταξύ δύο συνόλων σημείων-κλειδιών που έχουν ανιχνευτεί μέσω του SIFT. Η διαδικασία ξεκινά με την ανίχνευση και εξαγωγή των χαρακτηριστικών SIFT από τις δύο εικόνες χρησιμοποιώντας τις συναρτήσεις detectSIFTFeatures και extractFeatures. Αυτά τα χαρακτηριστικά αντιστοιχίζονται αρχικά μέσω της συνάρτησης matchFeatures, η οποία εφαρμόζει το Ratio Test για την αποφυγή ψευδών αντιστοιχιών, περιορίζοντας τις αντιστοιχίσεις στο 80% της αναλογίας μεταξύ του πρώτου και δεύτερου κοντινότερου γείτονα.

Μετά την αρχική αντιστοίχιση, εφαρμόζεται ο αλγόριθμος RANSAC μέσω της συνάρτησης estimateGeometricTransform2D για την εκτίμηση της γεωμετρικής μετασχηματιστικής σχέσης μεταξύ των δύο συνόλων σημείων. Σε αυτή την υλοποίηση, χρησιμοποιείται ο προβολικός μετασχηματισμός (projective transformation), ο οποίος είναι ικανός να μοντελοποιήσει πιο σύνθετες γεωμετρικές αλλαγές όπως η περιστροφή, η κλίση και η παραμόρφωση. Οι βασικές παράμετροι του RANSAC σε αυτή την εφαρμογή περιλαμβάνουν το MaxDistance, το οποίο ορίζεται σε 3 pixels, καθορίζοντας το μέγιστο επιτρεπτό απόσταση για ένα σημείο να θεωρηθεί inlier.

Ο RANSAC επιλέγει τυχαία υποσύνολα αντιστοιχιών και εκτιμά το μοντέλο μετασχηματισμού. Για κάθε μοντέλο, ελέγχει ποιες αντιστοιχίσεις συμφωνούν με το μοντέλο εντός του καθορισμένου κατωφλίου. Το μοντέλο με τον μεγαλύτερο αριθμό inliers θεωρείται το βέλτιστο και τα inliers χρησιμοποιούνται για την τελική γεωμετρική μετατροπή και την απεικόνιση των αντιστοιχιών.

Παρακάτω παρουσιάζεται το σχετικό τμήμα κώδικα που υλοποιεί την εκτίμηση του γεωμετρικού μετασχηματισμού με χρήση του RANSAC:

%% Εκτίμηση Geometric Transform με RANSAC  
fprintf('Εφαρμογή του αλγορίθμου RANSAC για εύρεση inliers...\n');  
[tform, inlierIdx] = estimateGeometricTransform2D(matchedPoints1, matchedPoints2, ...  
 'projective', 'MaxDistance', 3);  
  
inlierPoints1 = matchedPoints1(inlierIdx, :);  
inlierPoints2 = matchedPoints2(inlierIdx, :);  
  
fprintf('Αριθμός inliers μετά το RANSAC: %d\n', size(inlierPoints1,1));

Σε αυτό το τμήμα του κώδικα, η συνάρτηση estimateGeometricTransform2D λαμβάνει ως εισόδους τα αρχικά ζεύγη αντιστοιχιών (matchedPoints1 και matchedPoints2) και εκτελεί τον αλγόριθμο RANSAC για την εκτίμηση του προβολικού μετασχηματισμού. Το αποτέλεσμα tform περιέχει τις παραμέτρους του εκτιμημένου μετασχηματισμού, ενώ το inlierIdx είναι ένας λογικός πίνακας που υποδεικνύει ποιες αντιστοιχίσεις θεωρούνται inliers. Τα inlierPoints1 και inlierPoints2 αντιπροσωπεύουν τα υποσύνολα των αρχικών σημείων που συμφωνούν με το εκτιμημένο μοντέλο, δηλαδή τα αξιόπιστα ζεύγη αντιστοιχιών.

**ΑΣΚΗΣΗ 7**

Οι αλγόριθμοι **SIFT** (Scale-Invariant Feature Transform), **GLOH** (Gradient Location and Orientation Histogram) και **SURF** (Speeded Up Robust Features) αποτελούν τρεις από τις πιο διαδεδομένες μεθόδους για την ανίχνευση και περιγραφή σημείων-κλειδιών σε εικόνες. Παρά το γεγονός ότι όλοι αποσκοπούν στην εξαγωγή χαρακτηριστικών που είναι ανθεκτικά σε διάφορες μετασχηματιστικές παραμέτρους, παρουσιάζουν ουσιώδεις διαφορές ως προς την αρχιτεκτονική, την αποδοτικότητα και την απόδοσή τους.

**Ανίχνευση Σημείων-Κλειδιών** αποτελεί ένα από τα κύρια σημεία διαφοροποίησης μεταξύ αυτών των αλγορίθμων. Ο **SIFT** χρησιμοποιεί τη διαφορά Γκαουσιανών (Difference of Gaussian - DoG) για την ανίχνευση σημείων-κλειδιών σε διάφορες κλίμακες, εντοπίζοντας τοπικά ακρότατα στον χώρο κλίμακας και των εικόνων. Ο **GLOH**, ως επέκταση του SIFT, ακολουθεί την ίδια βασική διαδικασία ανίχνευσης μέσω DoG, αλλά προσφέρει πιο λεπτομερή περιγραφή των σημείων-κλειδιών χωρίς να τροποποιεί τη διαδικασία ανίχνευσης τους. Αντίθετα, ο **SURF** χρησιμοποιεί τον **Hessian Matrix** για την ανίχνευση σημείων-κλειδιών, επιτρέποντας την εντοπισμό σημείων με υψηλή καμπυλότητα, και εκμεταλλεύεται τις **Integral Images** για την ταχύτερη υπολογιστική διαδικασία, μειώνοντας σημαντικά το υπολογιστικό κόστος σε σύγκριση με τον SIFT.

Όσον αφορά την **Περιγραφή Σημείων-Κλειδιών**, ο **SIFT** παράγει descriptors 128 διαστάσεων που βασίζονται σε ιστογράμματα κατεύθυνσης και μέτρου κλίσης γύρω από κάθε σημείο-κλειδί, προσφέροντας ανθεκτικότητα σε περιστροφές, κλιμάκωση και φωτιστικές μεταβολές. Ο **GLOH** επεκτείνει αυτή την προσέγγιση χρησιμοποιώντας descriptors 512 διαστάσεων πριν την εφαρμογή της **Πρωταρχικής Συνιστώσας Ανάλυσης (Principal Component Analysis - PCA)** για τη μείωση των διαστάσεων σε 128, παρέχοντας πιο λεπτομερείς πληροφορίες για την τοπική περιοχή γύρω από το σημείο-κλειδί, βελτιώνοντας την αναγνωρισιμότητα αλλά αυξάνοντας και το υπολογιστικό κόστος. Αντίθετα, ο **SURF** δημιουργεί descriptors 64 διαστάσεων βασισμένους σε απαντήσεις Haar wavelet γύρω από κάθε σημείο-κλειδί, σχεδιασμένους για ταχεία υπολογιστική διαδικασία, διατηρώντας ταυτόχρονα υψηλή απόδοση στην αναγνώριση χαρακτηριστικών.

Η **Αποδοτικότητα και Υπολογιστικό Κόστος** διαφοροποιούνται επίσης σημαντικά μεταξύ των αλγορίθμων. Ο **SIFT**, αν και προσφέρει υψηλή ακρίβεια και ανθεκτικότητα, είναι σχετικά αργός λόγω της πολυπλοκότητας των υπολογισμών που απαιτεί, καθιστώντας τον λιγότερο κατάλληλο για εφαρμογές σε πραγματικό χρόνο. Ο **GLOH**, με την προσθήκη περισσότερων διαστάσεων στους descriptors, αυξάνει το υπολογιστικό κόστος τόσο στην εξαγωγή όσο και στην αντιστοίχιση των χαρακτηριστικών, προσφέροντας βελτιωμένη αναγνωρισιμότητα αλλά με επιπλέον πολυπλοκότητα. Αντίθετα, ο **SURF** είναι σχεδιασμένος για ταχύτητα και είναι πολύ πιο αποδοτικός από τον SIFT και τον GLOH, καθιστώντας τον ιδανικό για εφαρμογές που απαιτούν επεξεργασία σε πραγματικό χρόνο, χάρη στη χρήση των Integral Images και των απλοποιημένων descriptors.

Η **Ανθεκτικότητα σε Μετασχηματισμούς** είναι επίσης ένα κρίσιμο κριτήριο διαφοροποίησης. Ο **SIFT** παρέχει υψηλή ανθεκτικότητα σε περιστροφές, κλιμάκωση, αλλαγές φωτισμού και ελαφρές παραμορφώσεις, καθιστώντας τον κατάλληλο για ποικίλες εφαρμογές υπολογιστικής όρασης. Ο **GLOH** διατηρεί όλες τις ανθεκτικές ιδιότητες του SIFT και τις ενισχύει μέσω πιο λεπτομερών descriptors, προσφέροντας ακόμη μεγαλύτερη ανθεκτικότητα σε περίπλοκους μετασχηματισμούς. Ο **SURF**, ενώ επίσης προσφέρει ανθεκτικότητα σε περιστροφές και κλιμάκωση, μπορεί να έχει ελαφρώς χαμηλότερη ανθεκτικότητα σε σύγκριση με τον SIFT λόγω της απλούστευσης των descriptors, ωστόσο η ταχύτητά του τον καθιστά δημοφιλή επιλογή για εφαρμογές που απαιτούν γρήγορη επεξεργασία με αποδεκτή ακρίβεια.

Οι **Εφαρμογές και η Χρήση** των αλγορίθμων διαφέρουν ανάλογα με τις ανάγκες της κάθε εφαρμογής. Ο **SIFT** είναι ιδανικός για εφαρμογές που απαιτούν υψηλή ακρίβεια στην αναγνώριση και αντιστοίχιση χαρακτηριστικών, όπως η δημιουργία ψηφιακών καρτών, η αποτύπωση αντικειμένων και η αναγνώριση προσώπων. Ο **GLOH** χρησιμοποιείται κυρίως σε εφαρμογές όπου η υψηλή ακρίβεια στην αναγνώριση είναι προτεραιότητα και ο υπολογιστικός χρόνος δεν αποτελεί περιορισμό, όπως στην ανάλυση εικόνων υψηλής ανάλυσης και στην αναγνώριση αντικειμένων με λεπτομερή χαρακτηριστικά. Αντίθετα, ο **SURF** είναι κατάλληλος για εφαρμογές σε πραγματικό χρόνο, όπως η παρακολούθηση αντικειμένων σε βίντεο, η πλοήγηση ρομπότ και η αυξημένη πραγματικότητα, όπου η ταχύτητα είναι κρίσιμη παρά το ενδεχομένως μικρότερο κόστος ακρίβειας σε σύγκριση με τον SIFT.

Τέλος, η **Απόδοση και Ακρίβεια** διαφοροποιούνται μεταξύ των αλγορίθμων. Ο **SIFT** παρέχει υψηλή ακρίβεια στην αναγνώριση και αντιστοίχιση χαρακτηριστικών, αλλά με υψηλό υπολογιστικό κόστος, καθιστώντας τον κατάλληλο για εφαρμογές όπου η ακρίβεια υπερτερεί της ταχύτητας. Ο **GLOH** προσφέρει ακόμη μεγαλύτερη ακρίβεια στην αναγνώριση χαρακτηριστικών λόγω των πιο λεπτομερών descriptors, αλλά αυτό επιτυγχάνεται με επιπλέον υπολογιστικό κόστος. Ο **SURF**, αν και δεν επιτυγχάνει την ίδια ακρίβεια με τον SIFT και τον GLOH, η ταχύτητά του τον καθιστά ιδανικό για εφαρμογές που απαιτούν γρήγορη επεξεργασία με αποδεκτή ακρίβεια.

**ΑΣΚΗΣΗ 9**

**SIFT:** Για πολυπύρηνους επεξεργαστές, δείχνουμε ότι για μια εικόνα N×NN×N σε έναν επεξεργαστή με PP πυρήνες, η συνολική χρονική πολυπλοκότητα για την επεξεργασία της εικόνας είναι:

από **[1]**.

**GLOH:**

**SURF:**

**ΑΣΚΗΣΗ 10**

[1]<https://repository.lsu.edu/cgi/viewcontent.cgi?article=3720&context=gradschool_theses>

[2]<http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A33%20SIFT-GLOH-SURF.pdf>

[3] **Lowe, D. G.** (2004). *Distinctive image features from scale-invariant keypoints*. **International Journal of Computer Vision**, 60(2), 91–110.

[4] **Bay, H., Tuytelaars, T., & Van Gool, L.** (2006). *SURF: Speeded up robust features*. **Computer Vision and Image Understanding**, 110(3), 346–359.