

**HMMY 10/01/06**

**ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΕΥΡΕΣΗΣ ΑΛΛΑΓΩΝ ΚΑΙ  
ΕΝΤΟΠΙΣΜΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΕ VIDEO ΑΚΟΛΟΥΘΙΕΣ**

Ποθητός Βασίλειος ( wpothitos@yahoo.de)

ΕΠΙΤΡΟΠΗ: ΜΙΧΑΛΗΣ ΖΕΡΒΑΚΗΣ  
ΕΥΡΙΠΙΔΗΣ ΠΕΤΡΑΚΗΣ  
ΝΙΚΟΛΑΟΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

**ΣΥΝΟΨΗ**

Στην εργασία αυτή, τα θέματα περιλαμβάνουν τον εντοπισμό αλλαγών (change detection) και την παρακολούθηση αντικειμένων (object tracking) σε ομοιογενείς ακολουθίες video. Το πρώτο θέμα, αναφέρεται στη διαπίστωση διαφορών μεταξύ εικόνων του ίδιου video sequence (πχ μετακίνηση ανθρώπου), ενώ το 2<sup>ο</sup> στην πρόβλεψη σε πρότερο χρόνο της κίνησης ενός αντικειμένου στο sequence αυτό. Και οι δυο τεχνικές βρίσκουν άμεση εφαρμογή σε προβλήματα επεξεργασίας video, με κύριες εφαρμογές αυτές της επιτήρησης χώρων, των στρατιωτικών εφαρμογών εντοπισμού στόχων και radars, αλλά και στην βιοιατρική ( ογκολογικός έλεγχος, κατάσταση εγκεφάλου κτλ), στην φυσική έρευνα ( εύρος πετρελαιοκηλίδων), στην βιομηχανία της τηλεόρασης ( ειδικά εφέ, ντουμπλάρισμα χαρακτήρων κτλ) και πλήθος άλλων.

Σε πρώτη φάση ερευνούμε μια μεγάλη σειρά προσεγγίσεων για change detection, εξετάζοντας τις εφαρμογές και τις ανάγκες που πρέπει να ικανοποιεί ένας τέτοιος αποδοτικός αλγόριθμος. Σε δεύτερη φάση, επιλέγουμε βάσει συγκεκριμένων κριτηρίων τρεις από τους πλέον δημοφιλείς change detection αλγόριθμους και εξετάζουμε την απόδοσή τους σε εκτενή δοκιμές. Σκοπός μας είναι να επιλέξουμε τη «βέλτιστη προσέγγιση», την οποία θα χρησιμοποιήσουμε παρακάτω, σε ένα πλήρες σύστημα εύρεσης διαφορών και πρόβλεψης κίνησης που υλοποιούμε.

Το επόμενο στάδιο αφορά θέματα object tracking. Εξετάζουμε το ευρύτατα χρησιμοποιούμενο kalman filter και με τα δεδομένα της προηγούμενης φάσης, κάνουμε πράξη τη πρόβλεψη κίνησης ανθρώπων σε real world videos.

Σαν συμπέρασμα της όλης εργασίας, προτείνουμε ένα αρκετά ευσταθές σύστημα αντιμετώπισης τέτοιων προβλημάτων, που αποδίδει καλά σε πλήθος διαφορετικών καταστάσεων ενώ είναι αρκετά ευέλικτο για μελλοντική τροποποίηση και βελτίωση.

## **ΠΕΡΙΕΧΟΜΕΝΑ**

<b>-ΚΕΦ. 1 MONITONING/SURVEILLANCE SYSTEMS</b>	<b>Σελ.5</b>
<b>-ΚΕΦ. 2 CHANGE DETECTION/OBJECT SEGMENTATION</b>	
	<b>Σελ.8</b>
<b>2.1 ΕΙΣΑΓΩΓΗ</b>	<b>Σελ.9</b>
<b>2.2 ΑΠΑΙΤΗΣΕΙΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ</b>	<b>Σελ.9</b>
<b>2.3 THE 4-STEP MODELING</b>	<b>Σελ.12</b>
<b>2.3.1 Feature Extraction</b>	
<b>2.3.2 Feature Analysis</b>	
<b>2.3.3 Classification</b>	
<b>2.3.4 Post processing</b>	
<b>2.4 ΤΕΧΝΙΚΕΣ ΠΡΟΕΠΕΞΕΡΓΑΣΙΑΣ</b>	<b>Σελ.14</b>
<b>2.4.1 Feature Extraction</b>	
<b>2.4.1.1 Γεωμετρικοί Μετασχηματισμοί</b>	
<b>2.4.1.2 Μετασχηματισμοί Σε Επίπεδο Έντασης/Χρώματος</b>	
<b>A.Μετασχηματισμοί Χρωματικού Συστήματος</b>	
<b>B. Μετασχηματισμοί Μοντελοποίησης Της Συμπεριφοράς του</b>	
<b>Θορύβου</b>	
<b>2.4.1.3 Edge Detection Και Intensity</b>	
<b>2.4.1.4 Συμπεράσματα</b>	
<b>2.4.2 Feature Analysis</b>	
<b>2.4.2.1 Επιλογή Του Τύπου της Σύγκρισης Των Εικόνων</b>	
<b>2.4.2.2 Εικόνα Αναφοράς (Background)</b>	
<b>2.5 ΑΛΓΟΡΙΘΜΟΙ CHANGE DETECTION</b>	<b>Σελ.21</b>
<b>2.5.1 Κατηγοριοποίηση Αλγορίθμων</b>	
<b>2.5.1.1 Αλγόριθμοι Βασιζόμενοι Στην Διαφορά Και Κατωφλίωση Των</b>	
<b>Προς Σύγκριση Εικόνων</b>	
<b>2.5.1.2 Significance And Hypothesis Tests</b>	
<b>A. Significance Tests</b>	
<b>B. Likelihood Ratio Tests</b>	
<b>C. Συνδυαστικά Μοντέλα Πιθανοτήτων</b>	
<b>2.5.2 Predictive Models</b>	
<b>A. Χωρικά Μοντέλα</b>	
<b>B. Χρονικά (Temporal) Μοντέλα</b>	
<b>C. Shading Orientation Model</b>	
<b>2.5.3 Επίλογος – Ζητήματα Απόδοσης</b>	
<b>2.5.3.1 Change Mask Consistency</b>	
<b>2.5.3.2 Κριτήρια Επιδόσεων Change Detection</b>	
<b>A. Euler Number</b>	
<b>B. Region Compactness</b>	
<b>C. Eccentricity</b>	
<b>D. Labeling</b>	
<b>2.6 ΕΠΙΛΟΓΗ ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΓΚΕΚΡΙΜΕΝΩΝ ΑΛΓΟΡΙΘΜΩΝ</b>	<b>Σελ.37</b>
<b>2.6.1 Gaussian Noise Modeling</b>	
<b>2.6.2 Poisson Noise Modeling</b>	
<b>2.6.3 Euler Number</b>	
<b>2.6.4 Μέθοδος Aach</b>	

- A. Context-Adaptive Motion Detection
- B. Spatiotemporal Noise Detection
- C. Homomorphic Prefiltering

<b>- ΚΕΦ.3 ΑΛΓΟΡΙΘΜΟΙ OBJECT TRACKING</b>	<b>Σελ.47</b>
3.1 ΕΙΣΑΓΩΓΗ	Σελ.48
3.2 ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΛΟΓΙΚΗ	Σελ.49
3.3 ΦΙΛΤΡΑ KALMAN	Σελ.51
3.3.1 Ορισμός	
3.3.2 Στοιχεία Υπολογισμού Φίλτρων Kalman	
3.3.3 Γιατί Φίλτρα Kalman	
3.4 EKF- EXTENDED KALMAN FILTER	Σελ.55
<b>- ΚΕΦ.4 ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ CHANGE DETECTION</b>	<b>Σελ.56</b>
4.1 ΚΡΙΤΙΚΗ ΤΩΝ VIDEOS	Σελ.57
4.2 ΑΛΓΟΡΙΘΜΟΙ CHANGE DETECTION ΚΑΙ ΠΡΟ ΕΠΕΞΕΡΓΑΣΙΑ ΠΟΥ ΕΠΙΛΕΧΘΗΚΕ	Σελ.59
4.2.1 Επιλογή Προεπεξεργασίας	
4.2.2 Επιλογή Αλγορίθμων	
1) Gaussian Noise Modeling	
2) Poisson Noise Modeling	
3) Euler Number	
4.2.3 Μεταεπεξεργασία	
4.2.4 Επίλογος	
4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ	Σελ.63
4.3.1 Κριτήρια Ορισμού Των Tests	
1) Object Metric Του Euler Number	
2) Hole Metric Του Euler Number	
3) Euler Number	
4) Compactness	
5) Αριθμός Των Regions Όπως Προκύπτουν Από Labeling	
4.3.2 Ορισμός Των Tests	
4.3.3 ΑΠΟΤΕΛΕΣΜΑΤΑ	
4.3.3.1 Video 1	
4.3.3.2 Video 2	
4.3.4 Ανασκόπηση Δύσκολων Περιπτώσεων	
1) Flickering στο 1 <sup>ο</sup> Video	
2) Freezing στο 2 <sup>ο</sup> Video	
4.3.5 ΣΥΜΠΕΡΑΣΜΑΤΑ	

## **- ΚΕΦ.5 ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ OBJECT**

<b>TRACKING</b>	<b>Σελ.82</b>
<b>5.1 ΤΕΛΙΚΗ ΕΠΙΛΟΓΗ ΜΕΘΟΔΟΛΟΓΙΑΣ CHANGE DETECTION</b>	<b>Σελ.83</b>
1) Video 1	
2) Video 2	
<b>5.2 ΜΕΤΑΒΑΣΗ ΣΕ TRACKING- ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ</b>	<b>Σελ.85</b>
<b>5.2.1 Ορισμός Του Workload Για Το Tracking</b>	
1) Video 1	
2) Video 2	
<b>5.2.2 Ανάπτυξη Workload</b>	
<b>5.3 ΑΠΟΤΕΛΕΣΜΑΤΑ</b>	<b>Σελ.90</b>
1) Video 1	
2) Video 2	
<b>5.4 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΣΥΝΟΛΙΚΗ ΚΡΙΤΙΚΗ TRACKING</b>	<b>Σελ.92</b>
<b>- ΚΕΦ.6 ΕΠΙΛΟΓΟΣ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ</b>	<b>Σελ.93</b>
 <b><u>-APPENDICES</u></b>	 <b>Σελ.95</b>
<b>A. ERFC FUNCTION</b>	
1. Ορισμός	
2. Ανάλυση Σε Σειρά Taylor Και Επίλυση	
3. Υλοποίηση Στην Εργασία	
<b>B. NOISE ESTIMATION- LMEDS</b>	
<b>C. HOMOMORPHIC FILTERING</b>	
1) Spectral Domain Υλοποίηση	
2) Spatial Domain Υλοποίηση	
3) Υλοποίηση Με Lowpass Filter Στο Πραγματικό Επίπεδο	
<b>D. GRAYSCALE ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ</b>	
 <b><u>-ΒΙΒΛΙΟΓΡΑΦΙΑ</u></b>	 <b>Σελ.103</b>

## **ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ**

- Ορισμός Τον Surveillance Συστημάτων**
- Πρώτη Αναγνώριση Των Λειτουργιών Τους**

**ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΕΥΡΕΣΗΣ ΑΛΛΑΓΩΝ ΚΑΙ  
ΕΝΤΟΠΙΣΜΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΕ VIDEO ΑΚΟΛΟΥΘΙΕΣ**

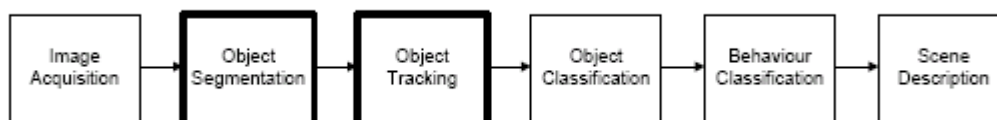
Η εύρεση αλλαγών μεταξύ εικόνων καθώς και ο εντοπισμός αντικειμένων που έχουν μετακινηθεί αποτελούν ένα σημαντικό πρόβλημα στο σύγχρονο πεδίο της οπτικής όρασης. Οι μέθοδοι που ασχολούνται με τα παραπάνω βρίσκουν εφαρμογή σε πλήθος πρακτικών προβλημάτων όπως η πλοήγηση, η επιτήρηση χώρων, ιατρικές εφαρμογές ( εντοπισμός όγκων κτλ) και άλλα. Στην παρούσα εργασία υλοποιούμε ένα τέτοιο σύστημα εύρεσης διαφορών / εντοπισμού αντικειμένων αφού πρώτα υποβάλουμε σε αρκετά διεξοδικές δοκιμές τρεις από τους πιο σημαντικούς αλγόριθμους εύρεσης αλλαγών (Gaussian modeling, Poisson modeling, Euler shape descriptor), ώστε να επιλέξουμε το καλύτερο από αυτούς, τουλάχιστον για τα video που έχουμε. Στη συνέχεια, μέσω kalman filtering υλοποιούμε και τον εντοπισμό αντικειμένων. Προκύπτουν τελικά μια σειρά σημαντικών συμπερασμάτων για τις διαδικασίες αυτές.

## 1.MONITORING/SURVEILLANCE SYSTEMS

Με το γενικό αυτό όρο περιγράφουμε **όλα** αυτά τα συστήματα στα οποία γίνεται εντοπισμός και περαιτέρω «επεξεργασία» μεταβολών μεταξύ εικόνων. Φυσικά **όλες** οι διαφορές μεταξύ των προς σύγκριση εικόνων τις πιο πολλές φορές δεν μας ενδιαφέρουν **αλλά προσπαθούμε να απομονώσουμε αυτές που απαιτεί η εφαρμογή** και να ακολουθήσουμε ανάλογη με τη περίπτωση επεξεργασία. Μερικά παραδείγματα

- i) Διαπίστωση εμφάνισης αντικειμένου (π.χ. αυτοκινήτου) σε μια σκηνή.
- ii) Διαπίστωση μεταβολής χλωρίδας (πχ πυρκαγιά)
- iii) Εύρεση πιθανών σφαλμάτων (πχ λάθος κολλήσεις σε ολοκληρωμένα) σε γραμμές παραγωγής
- iv) Αναγνώριση αντικειμένου (πχ ένας άνθρωπος ) σε μια σκηνή με πολλά ΟΜΟΙΑ αντικείμενα
- v) Πλοήγηση (πχ ρομπότ)
- κ.α.

Στα παραπάνω, οι απαιτήσεις **διαφοροποιούν τα συστήματα (και άρα δεν υπάρχει universal σύστημα, αλλά όλα είναι application depended)**, ωστόσο μπορούμε όλα να τα τυποποιήσουμε σε μια συγκεκριμένη ακολουθία:



**Image Acquisition :** Λαμβάνουμε τις εικόνες και έπειτα τις μετατρέπουμε σε μορφή κατάλληλη για την δουλειά που θέλουμε (πχ κάνουμε edge detection ή μετατρέπουμε σε grayscale/CIS Lab κτλ, κάνουμε differencing)

**Object Segmentation (Change Detection):** Απομονώνουμε το αντικείμενο(α)-μεταβολή (εξ) μεταξύ των εικόνων που θέλουμε να ερευνήσουμε (πιθανών κρατούμε και πληροφορία που δεν μας χρειάζεται, πχ θόρυβος)

**Object Tracking:** Επικεντρωνόμαστε **ακριβώς** στις μεταβολές/ αντικείμενα που μας ενδιαφέρουν, πχ βρίσκουμε που μετακινήθηκαν στη καινούργια σκηνή

Τα επόμενα στάδια αποτελούν περισσότερο εξειδικευμένα κάθε εφαρμογής και μπορούμε να τα κατατάξουμε στον ευρύτερο όρο του object understanding όπου προσπαθούμε να εξαγάγουμε ιδιότητες των μεταβολών που εντοπίσαμε (πχ είναι το αντικείμενο που εντοπίστηκε στη σκηνή αυτοκίνητο/ άνθρωπος/ όγκος /πυρκαγιά, είναι εμπόδιο που πρέπει να προσπελαστεί από ένα ρομπότ).

Η εργασία μας λοιπόν επικεντρώνεται στα τρία πρώτα στάδια, παρακάτω τα δύο πρώτα συγχωνεύονται και μιλάμε σε ποιο ειδικό βαθμό για τις απαιτήσεις, τα προβλήματα που παρουσιάζονται, τις εφαρμογές και τις μεθόδους που έχουν αναπτυχθεί τόσο για το Object segmentation/change detection όσο και για το Object tracking

## **ΚΕΦΑΛΑΙΟ 2 : CHANGE DETECTION/OBJECT SEGMENTATION**

- Εισαγωγή Στην Έννοια Του Change Detection, Απαιτήσεις**
- Τμηματοποίηση Της Διαδικασίας Σε Επιμέρους Στάδια**
- Προεπεξεργασία Και Αλγόριθμοι Προεπεξεργασίας, Συζήτηση Ως Προς Την Επίδραση Τους Στο Αποτέλεσμα**
- Ορισμός Της Σύγκρισης, Τύποι Σύγκρισης Και Επιλογές**
- Κύριο Τμήμα Αλγορίθμου, Κατηγοριοποίηση Αλγορίθμων, Συζήτηση Των Αποτελεσμάτων Σε Προηγούμενες Έρευνες**
- Απόδοση Και Ορισμός Μετρικών Αυτής**
- Συγκεκριμένοι Αλγόριθμοι (Poisson Noise Modeling, Gaussian Noise Modeling Euler Shape Descriptor, Aach Et Al Model)**

## **2. CHANGE DETECTION/OBJECT SEGMENTATION**

### **2.1 ΕΙΣΑΓΩΓΗ**



Η εύρεση περιοχών με αλλαγές μεταξύ εικόνων της ίδιας σκηνής, λαμβανομένων σε διαφορετικούς χρόνους είναι μια διαδικασία με ευρεία χρήση σε πολλές εφαρμογές και για διαφορετικούς σκοπούς. Εφαρμογές όπως video surveillance, remote sensing, ιατρικές εφαρμογές διάγνωσης, συστήματα driver assistance κα χρησιμοποιούν τέτοιους αλγόριθμους. Παρά το πολυδιάστατο των εφαρμογών αυτών, οι ερευνητές στο πεδίο χρησιμοποιούν μια σειρά κοινών ( ή τουλάχιστον όμοιων) τεχνικών και αλγορίθμων.

Η βασική ιδέα του προβλήματος είναι η εξής :

Έστω αρχικά ένα σύνολο εικόνων της ίδιας σκηνής που έχουν ληφθεί όμως σε διαφορετικούς χρόνους. Ο σκοπός μας είναι να προσδιορίσουμε των σύνολο των pixels σε αυτές που έχουν «αλλάζει σημαντικά» σε σχέση με μια άλλη εικόνα αναφοράς (που μπορεί να είναι η τελευταία εικόνα της σκηνής, η προηγούμενη κατά σειρά εικόνα στη διαδοχή της σκηνής κτλ ανάλογα πάντα την εφαρμογή). Αυτά τα pixels αποτελούν την μάσκα αλλαγής (change mask). Αυτή τελικά θα εμπεριέχει τα αντικείμενα που εμφανίστηκαν/ εξαφανίστηκαν στη σκηνή, κίνηση αντικειμένου σε σχέση με το background, αλλαγή σχήματος αντικειμένου κτλ (δηλ. χρήσιμη πληροφορία για την εφαρμογή) και αλλαγές λόγω κακής ποιότητας λήψης, κίνησης στο background (πχ μεταβολή φωτισμού) κτλ (δηλ. θόρυβος που πρέπει να αφαιρεθεί από την change mask. Αυτή είναι και η βασική απαίτηση ενός τέτοιου συστήματος, δηλ η change mask να μην εμπεριέχει τέτοιο «θόρυβο», δηλ. αλλαγές που δεν μας «ενδιαφέρουν». Αυτό μεταβάλλεται από εφαρμογή σε εφαρμογή (πολλές φορές ο «θόρυβος» για την μια είναι χρήσιμη πληροφορία για την άλλη), πράγμα που κάνει τους αλγόριθμους ισχυρά application depended και δύσκολο να συγκριθούν σε γενικό πλαίσιο.

Η προσέγγιση της change mask αποτελεί το πρώτο βήμα μιας σειράς άλλων εφαρμογών ( object tracking, semantic identification of objects κτλ).

## 2.2 ΑΠΑΙΤΗΣΕΙΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ

Για να ορίσουμε το πρόβλημα περισσότερο μαθηματικά, έστω  $\{ I_1, I_2 \dots I_M \}$  μια σειρά από εικόνες της ίδιας σκηνής όπου κάθε εικόνα αποτελείται από pixels με m τιμές εντάσεως ή k τιμές χρώματος. Τυπικά  $k=1$  (grayscale εικόνες) ή  $k=3$  (για παράδειγμα RGB color), αλλά και περισσότερες κάποιες φορές. Για παράδειγμα, πολυφασματικές εικόνες έχουν  $k = 10n$ , οι υπερφασματικές  $k=100n$ , όπου  $n=[0,1 \dots 10]$ . Τυπικά,  $m=2$  (satellite surveillance) ή  $m=3$  (αξονική τομογραφία, βιολογικές εικόνες μικροσκοπίου). Σε πολλές περιπτώσεις η σύγκριση γίνεται μεταξύ 2 μόνο εικόνων (σύγκριση εικόνων από δορυφόρο της ίδιας περιοχής της γης, μετά από μεγάλο χρονικό διάστημα για διαπίστωση μεταβολής της πανίδας κτλ) ή πολλών (πχ surveillance applications).

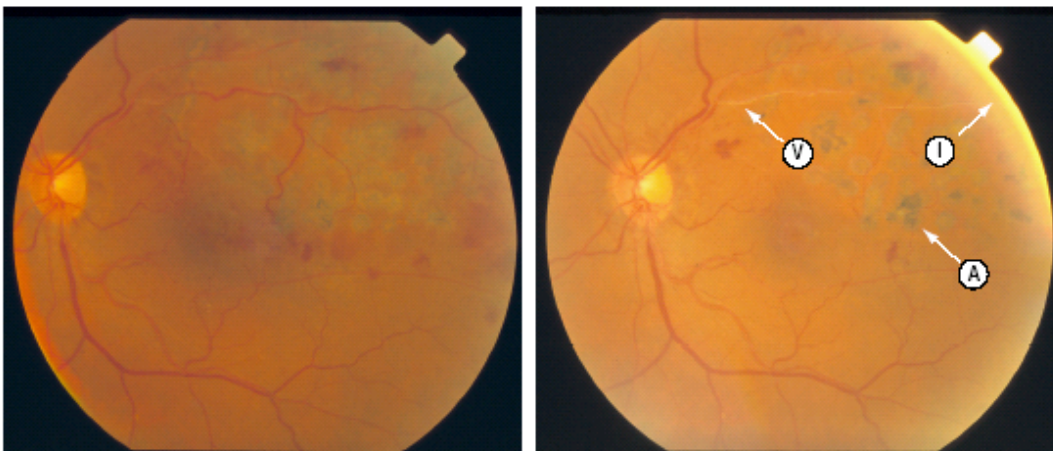
Η βασική ιδέα του change detection αλγόριθμου είναι να πάρει τις εικόνες που έχω ως είσοδο και να δημιουργήσει εικόνες πχ  $B(x,y)$  όπου σε αυτές

$B(x,y)=1$  αν έχω διαφορά στην εικόνα σε σχέση με μια εικόνα αναφοράς ή  $B(x,y)=0$  αν δεν έχω.

## EIKONA 1 CHANGES EXAMPLE Νο1



**ΕΙΚΟΝΑ 2 CHANGES EXAMPLE Νο2**



Στις παραπάνω εικόνες φαίνεται ακριβώς η πολυπλοκότητα του προβλήματος σε σχέση με τι ορίζουμε εμείς ως «Σημαντικές Αλλαγές» που ο αλγόριθμος πρέπει να διαπιστώσει.

Στην εικόνα 1, έχουμε τη σύγκριση δύο λήψεων σε διαφορετικό χρόνο, αντικειμένων πάνω σε ένα γραφείο. Οι αλλαγές μεταξύ των δύο εικόνων είναι:

- 1) Από ελαφρώς μετακίνηση της κάμερας
- 2) «Μη ευθεία» μετακίνηση της μπανάνας ( M )
- 3) Αντανάκλασεις φωτός ( S )
- 4) Μεταβολές στην ίδια την όψη των αντικειμένων ( A ), στίγματα στη μπανάνα

Στην εικόνα 2, έχουμε σύγκριση εικόνων του ίδιου αμφιβληστροειδούς σε διάστημα 6 μηνών, οι αλλαγές προέρχονται από

- 1) Μεταβολές στην αρτηριακή δομή ( V ) ( μια αρτηρία έχει νεκρώσει )
- 2) Αντανάκλασεις φωτός ( I )
- 3) Στίγματα στη δομή του αμφιβληστροειδούς ( A )

Κάποιες από τις παραπάνω αλλαγές είναι σημαντικές ( στην εικόνα 2 οι V,A ) και κάποιες όχι. Και στις δυο περιπτώσεις είναι δύσκολο να πούμε ποια ακριβώς είναι η change mask. Τα πάντα λοιπόν εξαρτώνται από την εφαρμογή που έχουμε και γενικός αλγόριθμος ΔΕΝ υπάρχει, ενώ είναι συζητήσιμο αν υπάρχει καν ο τέλειος αλγόριθμος

σε κάθε εφαρμογή (αυτός που δίνει όσο το δυνατόν πιο κοντά στο ιδανικό την change mask που απαιτείται).Ειδικά στο πεδίο εφαρμογής του video surveillance (όπου και η δική μας εργασία επικεντρώθηκε), γενικά δε θεωρούνται «σημαντικές» οι μεταβολές λόγω κίνησης της κάμερας, αντανakλάσεων κτλ. Μια σειρά δύσκολων καταστάσεων

- 1) Κίνηση αντικειμένου θεωρούμενα ως background (οχ κίνηση των φύλλων ενός δέντρου)
  - 2) Μεταβολές στη φωτεινότητα (σύγκριση εικόνας σε μέρα και νύχτα, στιγμιαίες μεταβολές φωτεινότητας κτλ)
  - 3) Αντικείμενα που **πρέπει** να εντοπιστούν αλλά δυσκολεύουν τον αλγόριθμο για μια σειρά λόγων ( camouflaged, μετατρέπονται σε ακίνητα για κάποιους λόγους κτλ)
  - 4) Σκιές των αντικειμένων που πρέπει να αφαιρεθούν
  - 5) Θόρυβος λόγω κακής ποιότητας λήψης που πρέπει να αφαιρεθεί
- Κτλ

Απ' την άλλη, σε άλλα πεδία εφαρμογών (ή σε περιπτώσεις με διαφορετικές απαιτήσεις, πχ αν θέλουμε να διαπιστώσουμε την αλλαγή από μέρα σε νύχτα ) κάποια από τα παραπάνω αποχτούν «σημασία» και πρέπει η αλλαγή να «διαπιστωθεί».

Γενικά, σε πρώτη προσέγγιση, ένας change detection αλγόριθμος πρέπει να είναι **ακριβής** στην εύρεση του περιγράμματος των αντικειμένων (spatial accuracy) και **χωρικά σταθερός** στο detection (temporal coherence), δηλ. να μην δίνει λχ ένα αντικείμενο στο ένα detection και στο άλλο να είναι «αλλοιωμένο».Ταυτόχρονα, πρέπει (ανάλογα τις απαιτήσεις της εφαρμογής) να είναι ικανός να διακρίνει μεταβολές «μικρής κλίμακας» (sensitivity) και να είναι σταθερός (robustness),δηλ. να μην επηρεάζεται από εξωγενείς παράγοντες (θόρυβος λόγω κακής λήψης, μεταβολές φωτεινότητας κτλ).

Spatial Accuracy, Temporal Coherence, Sensitivity, Robustness. Οι τέσσερις βασικές ιδιότητες που ο κάθε change detection αλγόριθμος πρέπει , όσο το δυνατόν γίνεται, να πληροί. Σε αυτές θα πρέπει να προστεθούν και ειδικές απαιτήσεις σε κάθε εφαρμογή, οχ περιορισμοί για «ακριβή αναπαράσταση ανατομίας/ γεωμετρίας του αντικειμένου» (ώστε να αναγνωριστούν συγκεκριμένα χαρακτηριστικά του οχ μύτη ,χέρια κτλ) ή και περιορισμοί άλλου τύπου (οχ υψηλή ταχύτητα επεξεργασίας ώστε να έχω μια real time εφαρμογή). Τα πάντα όμως εξαρτώνται από την εφαρμογή, η βαρύτητα κάθε ιδιότητας ορίζεται από τη φύση και τις δυσκολίες του προβλήματος. Συνεπώς, πριν από οτιδήποτε άλλο χρειαζόμαστε ένα πλήρη ορισμό των απαιτήσεων που έχουμε σε κάθε εφαρμογή και έπειτα να ορίσουμε την στρατηγική μας ανάλογα (οχ έχω indoor video surveillance, βρίσκω ανθρώπους σε κανονική κίνηση κτλ). Αυτό θα γίνει πιο εμφανές φθάνοντας στη δική μας υλοποίηση συστήματος και στην παρουσίαση των video αναφοράς μας.

Στο επόμενο κομμάτι, για να κατανοήσουμε περισσότερο τη δομή των αλγορίθμων αυτών, τη λειτουργία τους αλλά και για να μορφοποιήσουμε περισσότερο μαθηματικά το πρόβλημα, τμηματοποιούμε / μοντελοποιούμε την όλη διαδικασία του change detection σε 4 ανεξάρτητα στάδια.

## 2.3 THE 4-STEP MODELING

Η παρακάτω μοντελοποίηση [1] αποσκοπεί στην αποσαφήνιση και τμηματοποίηση της όλης διαδικασίας του change detection, ώστε να μπορέσουμε να την κατανοήσουμε σε βάθος και παρακάτω να παρουσιάσουμε αναλυτικά όλα τα στάδια αυτής.

Έστω λοιπόν  $I_i(x,y,n)$  μια ακολουθία εικόνων (της ίδιας σκηνής), όπου  $i$  = αριθμός εικόνων,  $(x,y,n)$  αναφέρεται στο pixel  $(x,y)$  και  $n$  οι διαστάσεις χρώματος/έντασης.

### 2.3.1 FEATURE EXTRACTION

Εδώ κάθε εικόνα της σκηνής μετατρέπεται σε μια κατάλληλη μορφή αναλόγως την εφαρμογή, το αποτέλεσμα, μια νέα σειρά εικόνων όπου σε αυτές θα εφαρμοστεί ο κυρίως αλγόριθμος. Επιλογή χρωματικού συστήματος (RGB, normalized RGB, CIS Lab, HSV κτλ), εύρος χρώματος ( 8-bit, 24-bit ), edge detection μετατροπές ,filtering ( homomorphic / median or average) κτλ εφαρμόζονται εδώ, ώστε αναλόγως τι ζητάμε από το σύστημα (πχ να μην επηρεάζεται από τα illumination variations) να έχουμε την κατάλληλη μορφή εικόνων για να το πετύχουμε. Η ουσιώδη χρήση των μετατροπών αυτών είναι να **κυρίως** (σε αναφορά τουλ. με τα συστήματα video surveillance) είναι να κάνουν τον αλγόριθμο να μην επηρεάζεται από τις μεταβολές στη φωτεινότητα και να μην την εντοπίσει σε μεταβολή.

Τέλος, εδώ κατηγοριοποιούνται και άλλες διεργασίες όπως το image registration, δηλ. αν η κάμερα κινείται κατά τη διάρκεια της λήψης να την «κλειδώσω» σε ένα κοινό σύστημα αναφοράς σε όλες τις εικόνες.

Στο 2.4 θα ασχοληθούμε περισσότερο με το feature extraction, εδώ απλώς παρουσιάζουμε κάθε στάδιο του change detection αλγόριθμου.

### 2.3.2 ΑΝΑΛΥΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ/FEATURE ANALYSIS

Μετά τη δημιουργία του κατάλληλου set εικόνων στο προηγούμενο στάδιο, σειρά έχει τώρα η σύγκριση αυτών με κάποια εικόνα αναφοράς . Ας συγκρατήσουμε τις δύο έννοιες ( Σύγκριση / Εικόνα Αναφοράς).

Η Εικόνα Αναφοράς μπορεί να είναι (ανάλογα την εφαρμογή) η προηγούμενη στη χρονική διαδοχή του video εικόνα ή μια εικόνα που θεωρούμε ως στατικό background της σκηνής. Και οι δυο επιλογές έχουν τα υπέρ / κατά τους [2]

Η χρήση του προηγούμενου frame ως Εικόνα Αναφοράς (δηλ. για κάθε frame διαφορά με το προηγούμενο του) χαρακτηρίζεται με τα εξής

- Εντοπίζοντας λιγότερα pixels αλλαγής (μόνο κίνηση) πλεονεκτεί στο ότι σε συστήματα επιτήρησης δίνει λιγότερα false alarms
- Όμως με αυτό τον τρόπο αντικείμενα που κινούνται αργά μπορεί να μην γίνουν αντιληπτά

Η χρήση στατικού background τώρα προϋποθέτει βέβαια την δημιουργία μιας τέτοιας εικόνας. Πέρα από την υπόθεση να πάρουμε μια εικόνα του video με «μηδενική» κίνηση , αν μπορούμε να έχουμε τέτοια βέβαια και χωρίς να ξεχνάμε την επίδραση των πολλών παραγόντων θορύβου , υπάρχουν μια σειρά απλές (όπως Η Average εικόνα από όλο το video) και πιο πολύπλοκες τεχνικές . Πλεονεκτεί στο ότι είναι πιο «ευαίσθητη» στις μεταβολές τεχνική και άρα μπορεί να δώσει αλλαγές για αντικείμενα που κινούνται πολύ αργά., αλλά είναι πιο ευάλωτη στα false alarms.

Η Σύγκριση τώρα δεν είναι τίποτα άλλο από την εφαρμογή μιας συνάρτησης μεταξύ του frame και της Εικόνας αναφοράς. Συνήθως αυτή είναι η απόλυτη τιμή της διαφοράς τους (ή το τετράγωνο της διαφοράς τους κτλ), αλλά υπάρχουν και άλλες μετρικές (likelihood ratio, image ratio κτλ).

Στο τέλος του σταδίου αυτού, μπορεί να λαμβάνουν χώρα και άλλες μετατροπές όπως πχ median filtering του αποτελέσματος της Σύγκρισης [3].

### 2.3.3 CLASSIFICATION

Εδώ λαμβάνει χώρα ο κυρίως αλγόριθμος change detection. Η βασική ιδέα είναι η εφαρμογή κατωφλίου (threshold) στις εικόνες που παρήχθησαν από το προηγούμενο στάδιο, ώστε τελικά να έχω εικόνες μόνο με τις αλλαγές (...τις «σημαντικές», κατά το δυνατόν). Το κατώφλι αυτό μπορεί να οριστεί εμπειρικά ή μέσω διάφορων , μεθόδων μοντελοποίησης (του θορύβου στην εικόνα κτλ). Μπορεί ακόμα να βελτιώνεται τοπικά στα regions της εικόνας .

Τέλος, το στάδιο αυτό δύναται να εμπεριέχει στη τελική του φάση διαδικασίες όπως πχ συνδυασμό αποτελέσματος δύο τεχνικών εύρεσης κατωφλίου (hysteresis thresholding) αλλά κυρίως (εφόσον δουλέψουμε με έγχρωμες – όχι grayscale- εικόνες ) τη διαδικασία συνδυασμού των αποτελεσμάτων του change detection αλγόριθμου σε κάθε χρωματική μπάντα ( πχ R G και B).

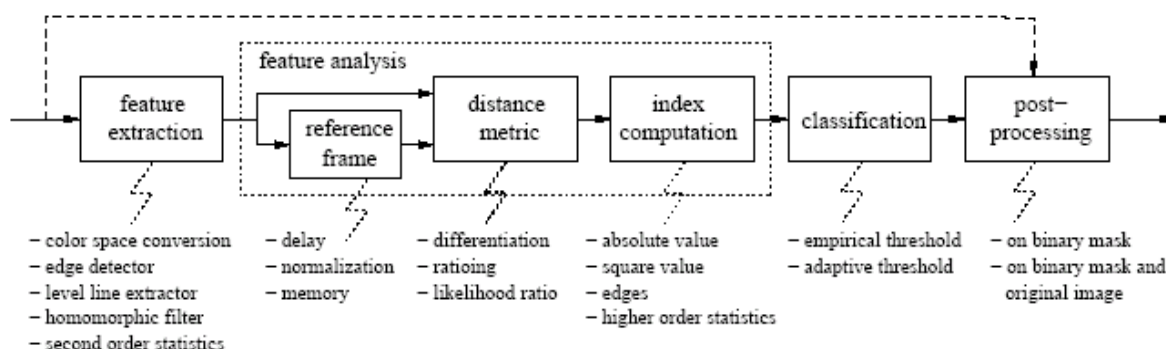
### 2.3.4 POST-PROCESSING

Από όσα είπαμε έως τώρα, το change detection ΣΠΑΝΙΑ αποτελεί αυτόνομο σύστημα αλλά συνήθως το πρώτο στάδιο που τα δεδομένα του προωθούνται σε ένα άλλο σύστημα ( για object tracking, object understanding κτλ). Στο post-processing λαμβάνει χώρα η περαιτέρω επεξεργασία των αποτελεσμάτων ώστε να είναι αυτό περισσότερο «Χρηστικό» για το επόμενο σύστημα αναγνώρισης, tracking κτλ.

Εδώ λοιπόν μπορεί να γίνει η αποθρομβοποίηση των εικόνων από artifacts όπως σκιές, θόρυβος κάμερας που δεν εξουδετερώθηκε τελείως, κομμάτια background που παρέμειναν στο τελικό αποτέλεσμα κτλ. Εδώ διάφοροι μορφολογικοί μετασχηματισμοί (dilations, erosions, συγκρίσεις με την αρχική εικόνα κτλ) χρησιμοποιούνται, ώστε τα **τελικά αποτελέσματα** του συνολικού change detection συστήματος να είναι κατάλληλα για την περαιτέρω επεξεργασία.

Συνολικά, μια πλήρης change detection διεργασία σχηματικά είναι ως παρακάτω

**EIKONA 3 THE 4-STEP MODEL OVERVIEW [1]**



Στα επόμενα κεφάλαια, παρουσιάζουμε πιο αναλυτικά τι γίνεται σε κάθε ένα από τα τέσσερα βασικά αυτά στάδια. Επαναλαμβάνουμε όμως ότι δίνουμε κυρίως βάση σε αλγόριθμους και τεχνικές video surveillance (δεν ασχολιόμαστε πχ αναλυτικά με τις βιοιατρικές μεθόδους ή με τις μεθόδους σε Satellite Surveillance).

Κάτι άλλο ακόμα, thresholding τεχνικές υπάρχουν πρακτικά 10-δες [7]. Εμείς επικεντρωνόμαστε σε αυτές που είναι δημοφιλής στο change detection.

## 2.4 ΤΕΧΝΙΚΕΣ ΠΡΟ ΕΠΕΞΕΡΓΑΣΙΑΣ

### 2.4.1 FEATURE EXTRACTION

#### 2.4.1.1 ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

Αλλαγές που οφείλονται σε κίνηση της κάμερας σχεδόν ποτέ δεν επιθυμούμε να διαγνωστούν ως «Πραγματικές αλλαγές» από τον change detection αλγόριθμο. Έτσι, σε περιπτώσεις που έχω κίνηση της κάμερας ένα σημαντικό βήμα είναι κατά πρώτον το image registration, η ευθυγράμμιση των εικόνων σε σχέση με ένα κοινό άξονα αναφοράς.

Όταν η κίνηση της κάμερας είναι μικρή και πχ ναι μεν έχω κίνηση της κάμερας αλλά γενικά το background είναι σταθερό, το registration μπορεί να γίνει με εφαρμογή χωρικών μετασχηματισμών και χρήση κριτηρίων όπως ομοιότητα περιοχών κτλ.

Η επιλογή του κατάλληλου χωρικού μετασχηματισμού και των κριτηρίων αυτού (χωρικά κριτήρια ή κριτήρια ομοιότητας έντασης κτλ) είναι σημαντική για το τελικό αποτέλεσμα (αν και στη δική μας εργασία παρακάτω δείχνουμε την «σταθερότητα» ορισμένων αλγορίθμων σε ελαφριά κίνηση της κάμερας κάτω από ορισμένες συνθήκες).

Το πιο σημαντικό πρόβλημα στο registration, είναι αυτή ακριβώς η επιλογή μετασχηματισμών με κριτήρια ομοιότητας έντασης ή χαρακτηριστικών (γεωμετρία αντικειμένων κτλ) ή συνδυασμός των παραπάνω. Συγκεκριμένα, όταν χρησιμοποιούμε κριτήρια χαρακτηριστικών των περιοχών, η ακρίβεια των χαρακτηριστικών αυτών καθορίζει σημαντικά περισσότερο την ακρίβεια του registration αλγόριθμου ( οι καλοί shape descriptors έχουν σημαντική ισχύ, όπως θα φανεί και στη δική μας δουλειά στο change detection). Σημαντικά λάθη στο change detection μπορεί να προκύψουν τοπικά στην εικόνα, ακόμα και αν ο registration αλγόριθμος δίνει συνολικά καλά αποτελέσματα.

Δεν ασχολούμαστε με το image registration διεξοδικά στην εργασία αυτή, αφού υποθέσαμε static camera. Στα tests βέβαια, ελαφρώς βρεθήκαμε σε camera movement, όμως αυτό ήταν μικρό, και **αποδείξαμε ότι με το σωστό preprocessing** οι αλγόριθμοι μας είναι αρκετά ανεπηρέαστοι.

#### 2.4.1.2 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ ΣΕ ΕΠΙΠΕΔΟ ΕΝΤΑΣΗΣ/ ΧΡΩΜΑΤΟΣ

**Ίσως**, το πιο σημαντικό preprocessing step ενός change detection αλγόριθμου. Αν υποθέσουμε σταθερή κάμερα (και στη δική μας εργασία αυτό κάναμε), η συντριπτική πλειονότητα των προβλημάτων προέρχεται πλέον από καταστάσεις όπως κακός ή μεταβαλλόμενος φωτισμός, αντικείμενα που «καμουφλάρονται» από το background κτλ. Καλή χρήση λοιπόν τέτοιων μετασχηματισμών, όπως αποδεικνύουμε, βελτιώνει σημαντικά τα αποτελέσματα. Παρακάτω, παρουσιάζουμε μια σειρά τέτοιων μετασχηματισμών, που βρήκαμε εν χρήσει στη βιβλιογραφία του change detection

## A. ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ ΧΡΩΜΑΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Ξεκινώντας εν γένει, από το απλό RGB, τριχρωματικό δηλ. , σύστημα απεικόνισης κάθε pixel , το να χρησιμοποιήσουμε ως αρχικό workload στο κυρίως αλγόριθμο αυτή καθ' αυτή τη πληροφορία όπως την λαμβάνουμε από την κάμερα, αποδεικνύεται **[11, αλλά και από μας, στο στάδιο των δοκιμών]** ανεπαρκές (για τους λόγους που εξηγήσαμε παραπάνω). Στη πράξη γενικά εφαρμόζονται τα εξής:

### 1) Normalized RGB

If  $K = R + G + B$

THEN  $R1 = R/K$

$G1 = G/K$

$B1 = B/K$

Είναι ιδιαίτερα δημοφιλές σε video surveillance **[20]**, πιο αποτελεσματικό από RGB και ιδιαίτερα σε αποκοπή των σκιών των αντικειμένων.

### 2) CIS Lab

Το σύστημα αυτό εξομαλύνει αρκετά τις χρωματικές ανομοιομορφίες στην ίδια περιοχή (region), πχ λόγω ανισομερής φωτεινότητας και αποκρύπτει περισσότερο τις «μικροδιαφορές» εντάσεων . Είναι δε πιο σταθερό στον ακανόνιστο θόρυβο (πχ λόγω κάμερας, είναι λογικό αφού οδηγεί σε πιο uniform περιοχές) **[11]**.

### 3) HSV

Εμφανίζει περισσότερη «ανεκτικότητα» (και άρα πιο ακριβές αλλά και πιο επιρρεπές στο θόρυβο) **[11]**.

Γενικά, οι μετασχηματισμοί αυτοί βελτιώνουν τη ποιότητα του αποτελέσματος, αλλά δεν είναι τόσο ισχυροί και τα αποτελέσματα τους διαφέρουν σημαντικά από εικόνα σε εικόνα, από περίπτωση σε περίπτωση video sequence. Θέλουμε κάτι πιο ισχυρό και με **γενικά** καλά αποτελέσματα.

## B. ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ ΤΗΣ ΣΥΜΠΕΡΙΦΟΡΑΣ ΤΟΥ ΦΩΤΙΣΜΟΥ

Εδώ, δεν μετασχηματίζουμε ομοιόμορφα όλες τις εικόνες , αλλά σε κάθε εικόνα προσπαθούμε να μοντελοποιήσουμε τη συμπεριφορά του φωτισμού. Δηλ. με την ισχύ βασικών μοντέλων της φυσικής προσπαθούμε να κάνουμε κάτι ανάλογο με το image registration για την κίνηση της κάμερας, μοντελοποίηση/ εξομάλυνση της «κίνησης»/ συμπεριφοράς του φωτισμού. Είναι λογικό και αποδεικνύεται από τις μετρήσεις **[14]** ότι αυτοί οι μετασχηματισμοί είναι ισχυρότεροι των παραπάνω (χρωματικά συστήματα HSV, CIS Lab κτλ), ιδιαίτερα σε περιπτώσεις απότομων αλλαγών του φωτισμού και μεταβολών φωτισμού που λαμβάνουν χώρα κατά τη διάρκεια της σκηνής.

## 1) INTENSITY NORMALIZATION

Εδώ, η ένταση στο Pixel στη μια εικόνα κανονικοποιείται σε σχέση με την ένταση στο ίδιο pixel με την Εικόνα Αναφοράς, ώστε οι δύο εικόνες να έχουν τελικά ίδια μέση τιμή και διασπορά της έντασης

$$\tilde{I}_2(x) = \frac{\sigma_1}{\sigma_2} \{I_2(x) - \mu_2\} + \mu_1$$

όπου  $\mu_i, \sigma_i$ , η μέση τιμή έντασης και η διασπορά αυτής σε κάθε εικόνα (αν έχω 3 μπάντες χρώματος και όχι grayscale, εφαρμόζω σε κάθε μπάντα πχ R G B) και  $\tilde{I}_2(x)$  η νέα normalized version της εικόνας προς σύγκριση με την εικόνα αναφοράς. Εναλλακτικά, και οι δύο εικόνες μπορούν να κανονικοποιηθούν ώστε να έχουν μηδενική μέση τιμή και διασπορά ίση με 1. Αυτό, μπορεί να βοηθήσει ώστε να επιλεχθούν thresholds στο change detection τα οποία είναι ανεξάρτητα από τις αρχικές εντάσεις στις εικόνες.

Ο τύπος μπορεί να εφαρμοστεί και τοπικά (ανά blocks) στις εικόνες, βελτιώνοντας την απόδοση του, με μειονέκτημα την δημιουργία block artifacts στα όρια των regions.

## 2) HOMOMORPHIC FILTERING

Σε πολλές περιπτώσεις, είναι δυνατό να υποθέσουμε ότι η ένταση σε κάθε pixel είναι ίση με

$$I(x) = I_l(x)I_o(x)$$

Δηλ. η συνολική ένταση είναι ίση με το γινόμενο της έντασης της προερχόμενης από τις πηγές φωτός  $I_l(x)$  στην εικόνα επί την αντανάκλαση  $I_o(x)$  των αντικειμένων [16]. Αυτό ονομάζεται shading model. Μόνο ο συντελεστής αντανάκλασης περιέχει πληροφορία για τα αντικείμενα στη σκηνή. Άρα, μια μορφή change detection ανεξάρτητη του φωτισμού και των μεταβολών του μπορεί να υλοποιηθεί με το να αφαιρέσουμε το συντελεστή των πηγών φωτός από την εικόνα.

Αν αυτός ο παράγοντας φωτός (Illumination Component) έχει χαμηλότερη χωρική συχνότητα από το συντελεστή αντανάκλασης ένα homomorphic filter μπορεί να χρησιμοποιηθεί ώστε να διαχωριστούν οι δυο αυτοί συντελεστές. Έτσι, υψώνοντας την συνολική ένταση σε λογάριθμο έχουμε

$$\ln I(X) = \ln I_l(x) + \ln I_o(x)$$

Και με τη χρήση ενός HP filter μπορούμε τώρα να πάρουμε πίσω το συντελεστή αντανάκλασης

$$\tilde{I}_o(x) = \exp\{F(\ln I(x))\}$$

Το homomorphic filtering είναι στη πράξη ένα πολύ ισχυρό εργαλείο στο change detection, καθώς το φυσικό μοντέλο που αναπαριστά προσεγγίζει πράγματι αρκετά καλά το πραγματικό φωτισμό στο μεγαλύτερο πλήθος των surveillance



προβλημάτων. Έτσι, καθίσταται απαραίτητο τμήμα πολλών change detection συστημάτων, ενώ είναι πολύ καλύτερο της απλής μεταβολής χρωματικού συστήματος (HSV, CIS Lab κτλ) [14]. Υπάρχουν μια σειρά σχεδιαστικών επιλογών στην υλοποίηση του, και επειδή είναι και η δική μας επιλογή για illumination-invariant change detection, αναλύουμε την υλοποίηση του σε ξεχωριστό παράρτημα στο τέλος της εργασίας μας.

### 3) ILLUMINATION MODELING

Στη συντριπτική πλειοψηφία των video, δηλ. σε video που η εικόνα είναι αυτή καθ' αυτή πχ δεν έχω underwater λήψη όπου τα ρεύματα του νερού επηρεάζουν την γεωμετρία των αντικειμένων, το Homomorphic Filtering είναι αρκετά ισχυρό εργαλείο για την αντιμετώπιση των illumination variations. Σε περιπτώσεις όμως που δεν ισχύει αυτό (μη LAMBERTIAN επιφάνειες), απαιτείται άλλου είδους προσέγγιση.

Υπάρχουν διάφορες μεθοδολογίες για την περίπτωση αυτή, αλλά και γενικά για την έκφραση ενός γενικευμένου illumination μοντέλου.

Οι Can και Singh [17] μοντελοποιούν το  $I_1(x)$  ως πολυώνυμο. Ο Negahdaripour [18] προτείνει ένα γενικευμένο μοντέλο

$$I_2(x) = M(x, y)I_1(x, y) + A(x, y)$$

Μεταξύ των  $I_1(x, y), I_2(x, y)$  ενδιάμεσων frames της ίδιας σκηνής.

Όπου  $M$ ,  $A$  ασυνεχή πολυώνυμα, από κει και πέρα υπάρχουν διάφορες προσεγγίσεις αυτών. Μια άλλη προσέγγιση (Hager et al), απαιτεί τη λήψη ενός αριθμού βασικών εικόνων σε διάφορες συνθήκες φωτισμού. Τότε η μεταβολή φωτισμού μεταξύ δύο διαδοχικών frames

$$I_2(x, y) = I_1(x, y) + \sum_{k=1}^K \alpha_k B_k(x, y)$$

Στη πράξη, αυτά τα μοντέλα χρησιμοποιούνται σπάνια, αφού σπάνια είναι και η περίπτωση μη LAMBERTIAN εικόνων.

### 4) ΕΠΙΠΛΕΟΝ ΕΙΔΙΚΕΣ ΠΕΡΙΠΤΩΣΕΙΣ

Υπάρχουν ακόμα διάφορα μοντέλα που αντιμετωπίζουν μερικές ιδιάζουσες, και σχετικά σπάνιες περιπτώσεις.

Οι Watanabe et al [19], εξετάζονται ακόμα περισσότερο τις περιπτώσεις **πολύ** απότομης μεταβολής του φωτισμού (πχ απότομη μεταβολή σε συννεφιά ή μετακίνηση σύννεφων και άρα συνεχώς μεταβαλλόμενος φωτισμός), κατέληξαν σε μια μεθοδολογία. Εδώ ξέροντας την θέση του ήλιου, σε σχέση με τα αντικείμενα της σκηνής, με μια μέθοδο ταιριάζουν κάποια patterns (κτιρίων συγκεκριμένα) στα υπάρχοντα αντικείμενα (κτίρια) της σκηνής, μπόρεσαν να αφαιρούν **ολικά** τις σκιές των κτιρίων, ώστε να μην εντοπιστούν από τον change detection αλγόριθμο ως αλλαγές.

Υπάρχουν ακόμα μεθοδολογίες αντιμετώπισης του λεγόμενου speckle noise, artifacts που παρουσιάζονται μεταξύ διαδοχικών εικόνων σε λήψης υπέρηχων.

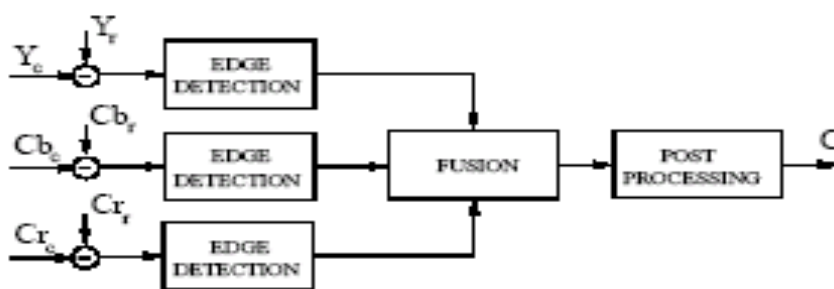
Προσεγγίσεις όπως multi-scale filtering, τοπικό spatial averaging και εφαρμόζονται εδώ.

### 2.4.1.3 EDGE DETECTION ΚΑΙ INTENSITY

Πάλι κυρίως για να δημιουργήσουμε ένα illumination independent σύστημα μπορούμε αντί τις intensity εικόνες να πάρουμε μια μετατροπή αυτών και να την δώσουμε ως workload στον change detection αλγόριθμο. Μετατροπές με χρήση pixels ratios, level lines, edge maps κτλ [9] έχουν χρησιμοποιηθεί προς αυτή την κατεύθυνση.

Η πιο πολυχρησιμοποιούμενη από αυτές τις τεχνικές είναι το edge detection, δηλ. η χρήση των edges αντί της κανονικής εικόνας. Αυτό γίνεται επειδή υποθέτουμε πως τα edges επηρεάζονται λιγότερο από το θόρυβο (illumination changes, camera noise κτλ), είναι κατά βάση shape descriptors, άρα δεν στηρίζονται τόσο σε intensity πληροφορία (χρώμα κτλ) που αλλοιώνεται ευκολότερα. Για παράδειγμα, Οι Cavallaro et al [9] προτείνουν μια μεθοδολογία Feature Extraction στηριζόμενη στα edges

### ΕΙΚΟΝΑ 4 CECD FEATURE EXTRACTION MODEL

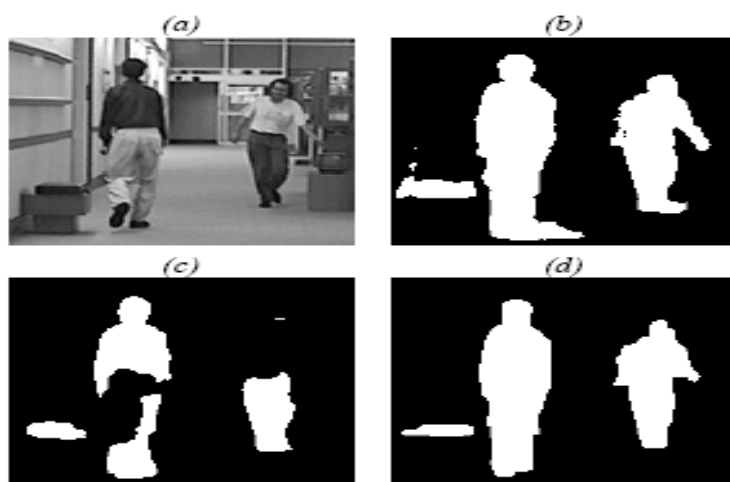


Δηλ. υλοποιούν το change detection με την παρακάτω αρχιτεκτονική 3 σταδίων

- 1) Edge Detection και στα 3 κανάλια χρώματος
- 2) Region filling με διαδοχικά dilations και erosions
- 3) Change Detection και Fusion των edge maps (3, ένα για κάθε κανάλι) σε ένα κοινό ( μέσω πχ averaging).

Πέρα από τις υπόλοιπες σχεδιαστικές επιλογές, αυτό που προκύπτει είναι η βελτίωση της επίδοσης μέσω edge detection όπως φαίνεται παρακάτω. Σημειώνουμε βέβαια πως οι συγγραφείς δεν αναφέρουν αρκετά για το κυρίως change detection αλγόριθμο που χρησιμοποίησαν ( μιλούν για εμπειρικά thresholds κτλ)

## EIKONA 5 CECD RESULTS



**A) ΑΡΧΙΚΗ ΕΙΚΟΝΑ B) EDGE MAP ΑΠΟ ΑΠΛΟ GRAYSCALE  
C) SHADING MODEL (HOMOMORPHIC Η' ΑΛΛΟ) D) CECD ΔΗΛ.  
COLOR EDGE DETECTION Ν' FUSION**

### 2.4.1.4 ΣΥΜΠΕΡΑΣΜΑ

Υπάρχουν και άλλες διαδικασίες που πιθανόν μπορούμε να δούμε στο Feature Extraction. Με χρήση διάφορων filters (averaging, blurring κτλ) μπορούμε να αφαιρέσουμε κάπως το θόρυβο και να βελτιώσουμε τη ποιότητα των frames του workload μας, να αποκόψουμε τμήματα που είναι εξ' αρχής μη χρήσιμα ( κάδρα κτλ), να χρησιμοποιήσουμε μοντέλα για αφαίρεση του καμουφλάζ από αντικείμενα. Δε κάνουμε όμως περισσότερο λόγο γι' αυτά, είναι γενικά προαιρετικά και η όλη μεθοδολογία τους ξεφεύγει από την σκοπό της εργασίας μας, που είναι η μελέτη πιο γενικευμένων μοντέλων για video surveillance.

Αυτό που θα πρέπει να σημειώσουμε κατά κύριο λόγο, είναι πως το Feature Extraction **δεν** πρέπει ποτέ να υποτιμάται, πράγμα που συμβαίνει πολλές φορές και αρκετοί ερευνητές δίνουν το κύριο όγκο της προσπάθειας τους στον change detection αλγόριθμο. Καλό Feature Extraction οδηγεί πάντα σε μεγάλη βελτίωση το αποτέλεσμα του change detection, και αυτό δεν το έδειξε μόνο η βιβλιογραφία που μελετήσαμε, αλλά θα φανεί και παρακάτω στα δικά μας tests .

Αυτό που πρέπει επιπλέον να προσεχθεί είναι ότι το Homomorphic Filter είναι ταυτόχρονα ένα γενικό και ισχυρό μοντέλο για την αντιμετώπιση των illumination variations. Γι' αυτό και στα δικά μας tests, εξ' αρχής προτιμάμε αυτό αντί πχ τη μελέτη της συμπεριφοράς της φωτεινότητας σε HSV, RGB , CIS Lab κτλ. Το προτιμάμε και από την υιοθέτηση ενός edge map model (πχ CECD) γιατί, απλώς στη συντριπτική πλειοψηφία της βιβλιογραφίας το edge detection δεν χρησιμοποιείτε τόσο πολύ, πιο πολύ η αντιπαράθεση γίνεται μεταξύ COLOR MODELS και HOMOMORPHIC. Η χρήση του ενός ( Homomorphic filtering ή CECD Feature Extraction ) καθιστά άχρηστο το άλλο, στη θεωρία και τα δυο οδηγούν σε

illumination invariant σύστημα, ώστε δεν συνίσταται ο συνδυασμός τους. Ακόμα, οι συγγραφείς του CECD **δεν** αναφέρουν αρκετά για το κυρίως change detection αλγόριθμο, και άρα προτιμούμε τη σιγουριά από την υπόλοιπη αναλυτικότερη βιβλιογραφία .

## 2.4.2 FEATURE ANALYSIS

Όπως είπαμε, εδώ συγκρίνουμε τις εικόνες από το προηγούμενο στάδιο και τροφοδοτούμε τα αποτελέσματα στο κύριο change detection αλγόριθμο. Πέραν όσα θέσαμε στην εισαγωγή για το feature analysis ( 2.3.2 ) μπορούμε να προσθέσουμε και κάποια επιπλέον θέματα

### 2.4.2.1 ΕΠΙΛΟΓΗ ΤΟΥ ΤΥΠΟΥ ΤΗΣ ΣΥΓΚΡΙΣΗΣ ΤΩΝ ΕΙΚΟΝΩΝ

Υπάρχουν αρκετές επιλογές για το πώς θα συγκρίνουμε τις εικόνες από το Feature Extraction.

Οι πιο εμφανείς επιλογές

1) Απλή αφαίρεση  $I = abs(I_2(x, y) - I_1(x, y))$

2) Τετράγωνο του  $I$

Αν χρησιμοποιούμε CIS Lab ή HSV οι διαφορές ορίζονται ως

i) CIS Lab [11]

$$D_{Lab} = \sqrt{(\Delta a)^2 + (\Delta b)^2} \quad \text{με } \Delta a = a_R - a_T$$

$$\Delta b = b_R - b_T$$

$a, b$  = color channels

ii) HSV [11]

$$D_{HSV} = \sqrt{(\Delta S_x)^2 + (\Delta S_y)^2} \quad \text{με } \Delta S_x = S_R \cos(H_R) - S_T \cos(H_T)$$

$$\Delta S_y = S_R \sin(H_R) - S_T \sin(H_T)$$

$H$  = hue  
 $S$  = saturation

(επειδή τα δύο συστήματα **δεν** είναι γραμμικά όπως το RGB )

και οι δυο επιλογές έχουν τα υπέρ και τα κατά τους. Η (2) δίνει περισσότερη πληροφορία (άρα πιο πολύ από τα αντικείμενα και περνά και πιο πολύ θόρυβο), η (1) αντιθέτως μειώνει την πληροφορία που περνά τελικά. Άρα, γενικά η επιλογή δεν είναι τόσο σημαντική, διαλέγουμε αναλόγως τι προτιμάμε. Στην εργασία μας επιλέγω την

(1), ακριβώς γιατί το τελικό αποτέλεσμα είναι πιο αποθρυβωποιημένο, υποθέτω ( και ορθώς τελικά) πως η region πληροφορία που θα πάρω είναι αρκετή.

#### 2.4.2.2 ΕΙΚΟΝΑ ΑΝΑΦΟΡΑΣ (BACKGROUND)

Όταν η σύγκριση γίνεται μεταξύ του frame και του προηγούμενου του ασφαλώς και η Εικόνα Αναφοράς κάθε frame είναι το προηγούμενο του. Όταν κάνουμε σύγκριση μεταξύ του frame και κάποιου υποτιθέμενου background ερχόμαστε αντιμέτωποι με μια σειρά προβλημάτων που έχουν να κάνουν με τον υπολογισμό αυτού.

Κατά πρώτον, μπορούμε να επιλέξουμε για background μια εικόνα του video στην οποία δεν έχω κινούμενα αντικείμενα. Αυτή η επιλογή δεν είναι πάντα δυνατή (έχω κίνηση σε όλο το video, δεν μπορώ να βρω εικόνα εντελώς κενή από αντικείμενα κτλ), ενώ εμπεριέχει μια σειρά προβλήματα, με σημαντικότερο τη διαφορά φωτεινότητας μεταξύ frame-background (λόγω χρονικής απόστασης κτλ). Το πρόβλημα αυτό μπορεί να λυθεί με χρήση homomorphic στο background image και άλλων συναφών μεθόδων.

Στη περίπτωση που αυτό δεν μπορεί να γίνει, λχ στο video έχω συνεχή κίνηση, πρέπει να υπολογίσω ένα background image με κάποιο τρόπο. Αυτό μπορεί να γίνει με μια σειρά μεθόδων, με τελικό αποτέλεσμα είτε ένα background image ή ένα set backgrounds για κάθε time slot/frame του video.

Σημαντική δουλειά έχει γίνει στο background maintenance [8, και αλλού] .

Οι μέθοδοι που μπορούν να χρησιμοποιηθούν ποικίλουν. Μπορούμε λχ να κάνουμε απλά averaging των εικόνων που έχουμε, αυτό πιθανών να εμπεριέχει κάποια προβλήματα, όπως ένα αντικείμενο να βρίσκεται για πολύ ώρα στη σκηνή, συνεπώς να τοποθετηθεί στο Background. Υπάρχουν και πιο πολύπλοκες τεχνικές. Γενικά, τα προβλήματα που αντιμετωπίζουμε εδώ είναι ανάλογα με τα γενικά προβλήματα ενός change detection αλγορίθμου ( illumination changes, αντικείμενα με μικρή κίνηση, αντικείμενα που ενώ είναι ακίνητα ξαφνικά αρχίζουν να κινούνται, καμουφλάζ κτλ).

Ένα σημαντικό reference στη διαδικασία αυτή αποτελεί το [8]. Εδώ οι Toyama et al προτείνουν έναν υψηλής απόδοσης αλγόριθμο ( Wallflower ), παρουσιάζουν ένα διεξοδικό test μιας σειράς background maintenance μεθόδων, αποκρυσταλλώνουν μια σειρά προβλημάτων που υπάρχουν και τις προϋποθέσεις που ένας τέτοιος αλγόριθμος πρέπει να πλήρη.

Εμείς, δεν χρειάστηκε να ασχοληθούμε ιδιαίτερα με το στάδιο αυτό (απλώς κάναμε image averaging όπου χρειαστήκαμε να υπολογίσουμε background), και αυτό γιατί τα αποτελέσματα αυτού ήταν αρκετά καλά (όπως θα φανούν από τις μετρήσεις μας).

### 2.5 ΑΛΓΟΡΙΘΜΟΙ CHANGE DETECTION

Φθάνοντας στο κυρίως μέρος του change detection αλγόριθμου, έχουμε απορρίψει το περιττό, δηλ. το background και πλέον επιθυμούμε να ασχοληθούμε με τις διαφορές σε επίπεδο αντικειμένων , είτε ως διαπίστωση του ίχνους κίνηση τους, είτε ως διαπίστωση της αλλαγής των χαρακτηριστικών τους κτλ. Γενικά, οι αλγόριθμοι που υπάρχουν εδώ είναι πολύ και ποικίλουν ως προς τη μαθηματική τους βάση. Από απλούς threshold decision αλγόριθμους που συγκρίνουν τα χαρακτηριστικά σε μια γειτονιά του pixel [Kolmogorov-Smyrnov test 10 κα], συστήματα στοχαστικών μετρήσεων στηριζόμενα σε μοντέλα θορύβου ή σε shape

descriptors [Euler method, Poisson και Normal modeling κα 5] ως τη χρήση περισσότερο εξεζητημένων μαθηματικών μοντέλων όπως είναι οι αλυσίδες Markov και τα νευρωνικά δίκτυα [21], ενώ και υβριδικές μέθοδοι ( συνδυασμός αποτελεσμάτων μεταξύ διαφορετικών από τις παραπάνω μεθόδους ) είναι αρκετά δημοφιλείς.

## 2.5.1 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ

Γενικά, το να προσπαθήσουμε κατηγοριοποίηση των change detection αλγορίθμων φαντάζει αρκετά δύσκολο, όσο δύσκολο και πολυδιάστατο είναι το πρόβλημα του change detection. Κάθε αλγόριθμος ακολουθεί μια φιλοσοφία η οποία λειτουργεί καλά ανάλογα το detection πρόβλημα που επιθυμεί να επιλύσει. Με απλά παραδείγματα, οι αλγόριθμοι που χρησιμοποιούνται σε medical εφαρμογές δεν λειτουργούν το ίδιο καλά με τους ανάλογους για video surveillance , γιατί πολύ απλά είναι διαφορετικές οι απαιτήσεις των προβλημάτων αυτών. Μια κατηγοριοποίηση, τουλάχιστον με έμφαση και προσανατολισμό το video surveillance δίνει:

### 2.5.1.1 ΑΛΓΟΡΙΘΜΟΙ ΒΑΣΙΖΟΜΕΝΟΙ ΣΤΗΝ ΔΙΑΦΟΡΑ ΤΩΝ ΠΡΟΣ ΣΥΓΚΡΙΣΗ FRAMES ΚΑΙ ΕΥΡΕΣΗ THRESHOLD ΜΕΣΩ ΑΥΤΗΣ

Αποτελούν την πλειονότητα των αλγορίθμων που υπάρχουν. Στη βασική τους ιδέα οι αλγόριθμοι αυτοί είναι σχετικά απλοί, αλλά με κάποιες παραλλαγές ( χρήση shape descriptors όπως euler number, επιλογή κατωφλίου μέσω spatial-temporal modeling κτλ) γίνονται αρκετά ισχυροί, όπως θα δείξουμε παρακάτω.

Κάθε τέτοιος αλγόριθμος, όπως για άλλη μια φορά επαναλαμβάνουμε, βασίζεται στη διαφορά ( με τους διάφορους τρόπους που ορίσαμε παραπάνω) μεταξύ των δύο εικόνων.

Difference  $D(x, y) = I_2(x, y) - I_1(x, y)$

και έπειτα μέσω κάποιου threshold  $\tau$  έχω  $B(x, y) = 1$  , αν  $D > \tau$   
0, αλλιού

ως τελική change mask.

Οι αλγόριθμοι αυτοί ομαδοποιούνται κάτω από τον γενικό όρο «Simple Differencing» και από εκεί και πέρα οι μέθοδοι εύρεσης του κατωφλίου είναι αρκετές. Μπορούμε πάντα να το θέσουμε εμπειρικά, αλλά υπάρχουν αρκετές άλλες επιλογές. Ο Rosin [5] παρουσιάζει μια σειρά μεθόδων που χρησιμοποιούμε σήμερα, και είναι αυτές που χρησιμοποιήσαμε και εμείς και θα αναλύσουμε λεπτομερέστερα παρακάτω, οι περισσότερες εκ των οποίων ξεφεύγουν από την απλή «εμπειρική» λογική, ακολουθούν μια σειρά υποθέσεων για τη συμπεριφορά του θορύβου και μέσω μοντελοποίησης της συμπεριφοράς του ευρίσκουμε πραγματικά ικανοποιητικά αποτελέσματα. Μια ακόμα παραλλαγή της βασικής λογικής είναι η λεγόμενη change vector analysis, συνήθως χρησιμοποιούμενη σε multispectral εικόνες. Σε αυτή, ένας πίνακας χαρακτηριστικών (feature vector) δημιουργείται για κάθε pixel της εικόνας, για τα πολλά φασματικά αυτά πεδία (multispectral). Το modulus της διαφοράς των δύο feature vector σε κάθε pixel δίνει τις τιμές της εικόνας διαφοράς (  $D(x)$  ). Αντί της διαφοράς μπορούμε να χρησιμοποιήσουμε ratioing (δηλ. αντί της διαφοράς η ένταση σε κάθε pixel συνεισφέρει στη τελική τιμή της  $D(x)$  με κάποιο ποσοστό.

Όπως είπαμε, υπάρχουν πολύ τρόποι επιλογής κατωφλίου. Η βασική μαθηματική λογική που διέπουν τους περισσότερους αλγόριθμους αυτούς αναλύεται στο αμέσως επόμενο κομμάτι

### 2.5.1.2 SIGNIFICANCE AND HYPOTHESIS TEST

Ο κανόνας επιλογής κατωφλίου στους περισσότερους αλγόριθμους μαθηματικοποιήτε ως ένας στατιστικός έλεγχος υπόθεσης (statistical hypothesis test). Η απόφαση αλλαγή/ μη αλλαγή σε ένα πίξελ  $\chi$  αντιστοιχεί στην επιλογή μιας από τις δυο «αντίθετες» υποθέσεις : την υπόθεση «μηδέν»  $H_0$  (null hypothesis = no-change) και της αντίθετης της  $H_1$  (alternative hypothesis = change).

Το σετ εικόνων που συγκρίνω κάθε φορά  $I_1(x, y), I_2(x, y)$  μπορεί να ειπωθεί ως ένας τυχαίος πίνακας. Η γνώση των pdf

$$p(I_1(x, y), I_2(x, y) | H_0) \text{ και } p(I_1(x, y), I_2(x, y) | H_1)$$

(δηλ. των συναρτήσεων πιθανότητας αλλαγής/ μη αλλαγής μας επιτρέπει να επιλέξουμε την υπόθεση που περιγράφει καλύτερα την αλλαγή έντασης στο  $I_2(x, y)$  σε σχέση με το  $I_1(x, y)$  με τη χρήση διαφόρων μοντελοποιήσεων (πχ Aach method, Gaussian κτλ [5]) με χρήση κλασικής θεωρίας πιθανοτήτων (παρακάτω, στην περαιτέρω ανάλυση κάποιων αλγορίθμων θα δούμε με παράδειγμα πως προκύπτει έτσι η επιλογή του κατωφλίου στη περίπτωση της μοντελοποίησης με GAUSSIAN και στη παραπλήσια μέθοδο του Aach).

Ακόμα, αφού οι σημαντικές αλλαγές τείνουν να εμφανίζονται κατά regions από pixels και όχι μεμονωμένα ( εντοπίζουμε αντικείμενα, όχι «κουκίδες»), είναι συνηθισμένη τακτική για τον εκάστοτε change detection αλγόριθμο που στηρίζεται στη θεωρία αυτή να «κρίνει» το αν έχω αλλαγή/ μη αλλαγή για το pixel  $x$  με βάση κριτήρια προερχόμενα από τη γειτονιά του pixel. Η προσέγγιση αυτή ονομάζεται «Geo-pixel» (τα κριτήρια επιλογής για το pixel βασίζονται στις συμπεριφορά της region και όχι μόνο στο pixel αυτό καθαυτό). Φυσικά πάντα μπορούμε να έχουμε και το δεύτερο, απόφαση για το pixel μόνο από αυτό.

Ανάλογα και παραπλήσια με το προηγούμενο, μπορούμε να εφαρμόσουμε το κριτήριο αλλαγής/ μη αλλαγής είτε σε κάθε pixel ξεχωριστά είτε σε regions (πχ 5x5 ).

Ένα παράδειγμα για να αποσαφηνιστεί το παραπάνω, έστω πχ μια region 3x3 pixels

- i) Το να θέσω στο pixel τη μέση τιμή της region είναι πράξη που στηρίζεται στη συμπεριφορά της γειτονιάς αλλά εφαρμόζεται στο pixel
- ii) Το να θέσω τη μέση τιμή σε όλη τη region (ή πχ τη τρέχουσα τιμή του pixel) είναι μια εφαρμογή μεθόδου σε όλη τη region.

Το να εφαρμόσουμε λοιπόν block per block testing ( δηλ. εφαρμογή του αλγορίθμου σε κάθε block, ανεξάρτητα της δομής της εικόνας ) αντί του pixel προς pixel, πχ να πάρουμε την εικόνα και να πούμε πως έχουμε να κάνουμε με  $N$  ξεχωριστές εικόνες, έχει το μειονέκτημα του ότι μη λαμβάνοντας πλέον υπόψη τη δομή κτλ των αντικειμένων που υπάρχουν θα πάρουμε χειρότερο αποτέλεσμα και block artifacts μεταξύ των region. Από την άλλη, θα μειώσουμε αισθητά το χρόνο επεξεργασίας ( ιδιαίτερα αν έχουμε στη διάθεση μας ένα καταναμεμημένο σύστημα. Τελικά, δεν υπάρχει ο Χρυσός Κανόνας επιλογής ούτε και εδώ , δουλεύουμε ανάλογα

τι θέλουμε και τι έχουμε. Παρακάτω, παρουσιάζουμε κάποια ευρέως χρησιμοποιούμενα γενικά κριτήρια των  $H_1 / H_0$  για την εύρεση του κατωφλίου. Η κατανόηση τους θα μας φανεί χρήσιμη όταν αναλύσουμε διεξοδικά τους αλγορίθμους που στηρίζονται σε αυτά.

### A) Significance Tests

Ο χαρακτηρισμός της  $H_0$  (null hypothesis = no change) είναι συνήθως άμεσος και ευθύς, αφού με την απουσία αλλαγών η όποια διαφορά μεταξύ  $I_1(x, y)$  και  $I_2(x, y)$  Pixels οφείλεται μόνο στο θόρυβο. Ένα significance test στο difference image μπορεί να εφαρμοστεί ώστε για ένα αποδεκτό ποσοστό λαθών

$\alpha$  = λάθος καταχωρημένα pixels ως αλλαγές ή μη αλλαγές / Σύνολο pixels στην εικόνα

τότε το τεστ ορίζεται ως

$$S(x) = p(D(x)) | H_0, \text{ όπου } \alpha > \tau, H_0 \\ \text{αλλιώς } H_1$$

η σημειολογία και οι συμβολισμοί θα μας γίνουν πιο κατανοητοί στην αναλυτική παρουσίαση αλγορίθμων, χοντρικά τώρα μπορούμε να πούμε πως  $\alpha$  είναι μια μέση τιμή του  $S(x)$  = πιθανότητα δεδομένου πως δεν είχα αλλαγή να καταχωρηθεί αυτό ως αλλαγή για κατώφλι  $\tau$ , δηλ. να είναι θόρυβος. Ο κάθε ερευνητής εστιάζει πλέον στην μοντελοποίηση του  $S$  ώστε να έχει τα καλύτερα αποτελέσματα. Ο Aach [14] υποθέτει κανονική κατανομή, άλλοι poisson κτλ. Λίγα λόγια για την περίπτωση της γκαουσιανής, υποθέτουμε πως ο θόρυβος στην εικόνα ακολουθεί την γκαουσιανή/ κανονική κατανομή με διασπορά  $\sigma_0$  και μέση τιμή 0. Υπάρχουν αρκετές μεθοδολογίες για την εύρεση μιας τιμής στο  $\sigma_0$  [6 **αλλά λέμε και εμείς κάποια πράγματα στο παράρτημα**]. Σημαντικότερες προσεγγίσεις για τη διασπορά του θορύβου βρίσκονται σε αυτό το παράρτημα αλλά υπάρχουν και άλλοι τρόποι πχ από όλο το video βρίσκω περιοχές που δεν έχουν αλλάξει πολύ και άρα η διαφορά εκεί είναι μόνο θόρυβος.

Στην περίπτωση γενικά της κανονικής κατανομής η PDF είναι

$$p(D(x)) | H_0 = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left\{-\frac{D^2(x)}{2\sigma_0^2}\right\}$$

Στην περίπτωση που τμηματοποιήσουμε την εικόνα σε regions και εργαστούμε ξεχωριστά σε αυτά η μετατροπή των τύπων της pdf είναι εύκολη. Υποθέτοντας τώρα ότι ο θόρυβος είναι ανεξάρτητος από block σε block και κατανεμημένος ομοιόμορφα στην εικόνα τότε πχ ο τύπος της Gaussian του Aach γίνεται

$$p(D(x)) | H_0 = \left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^N \exp\left\{-\frac{\sum_{y \in \Omega_z} D^2(y)}{2\sigma_0^2}\right\} = \left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^N \exp\left\{-\frac{G(x)}{2}\right\}$$



Δηλ. το exp του συνόλου των Gaussian όλων των regions μεταφράζεται, όπως θα δούμε στον ίδιο των αλγόριθμο σε μια και μοναδική κατανομή  $x^2$  N βαθμών ελευθερίας με το  $N = f(\text{regions})$ , και από τους πίνακες της κατανομής μπορούμε να βρούμε το threshold για δοθέν α.

## B) LIKELIHOOD RATIO TESTS

Όπως είδαμε στο προηγούμενο, η μεθοδολογία υπολογισμού μέσω της  $H_0$  είναι εύκολη, από τη στιγμή που κάνουμε κάποιες υποθέσεις για την πιθανοτική συμπεριφορά του θορύβου. Αυτό είναι λογικό διότι ο θόρυβος μπορεί να προσεγγιστεί ευκολότερα, δηλ. να έχω μια a priori καλή πρόβλεψη αυτού.

Δηλ. με ένα αισθητικό παράδειγμα, αν έχω K εικόνες, αν σε αυτές μια περιοχή δεν αλλάζει μπορώ (με τις υποθέσεις για τη συμπεριφορά του θορύβου και την ομοιομορφία του) να έχω ένα σχετικά αξιόπιστο υπολογισμό του σ που να ισχύει για την K+1.

Όμως το αν αλλάξει μια περιοχή δεν μπορούμε να το ξέρουμε (η τουλ. με την λογική που δουλεύουμε στο πεδίο αυτό, να ξέρουμε αν αλλάξει με μεγάλη πιθανότητα) a priori. Συνεπώς το να εργαστούμε με το  $H_1$  και να προσπαθήσουμε να μοντελοποιήσουμε τη συμπεριφορά των αλλαγών αντί του θορύβου είναι πολύ δυσκολότερο. Γιατί, όπως είπαμε παραπάνω, οι παρατήρηση της συμπεριφοράς των αλλαγών δεν μπορεί να γίνει a priori ή τουλ. να μοντελοποιηθεί σε ένα πιθανοτικό μοντέλο, είναι εντελώς (...η σχεδόν τουλάχιστον) απρόβλεπτη.

Αν υποθέσουμε όμως ότι και τα δύο μοντέλα ( $H_0$  = συμπεριφορά θορύβου,  $H_1$  = συμπεριφορά των αλλαγών), συνδυάζοντας τις δυο pdf (probability density function) έχω το likelihood ratio

$$L(x) = \frac{p(D(x) | H_1)}{p(D(x) | H_0)} \text{ (δηλ. ποσοστό αλλαγών / ποσοστό λάθους στην εικόνα)}$$

Το likelihood ratio ευρίσκεται σε άμεση συνάρτηση με το threshold τότε

$$\tau = \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})}$$

όπου  $P(H_i)$  [ $i = 0, 1$ ] οι a priori πιθανότητες των  $H$  και  $C_{ij}$  παράμετροι κόστους όπου  $i$  = επιλογή που έχω κάνει για τη θέση (αλλαγή ή όχι)  $j$  η πραγματικότητα (αλλαγή ή όχι). Συγκεκριμένα, ο  $C_{10}$  αντικατοπτρίζει τα false alarms (έθεσα σε μια θέση αλλαγή ενώ δεν έχω) και ο  $C_{01}$  το λάθος συμπέρασμα αν σε μια θέση έχω αλλαγή αλλά δεν τη δώσει ο αλγόριθμος.

Οι Aach et al [15] επέκτειναν τον αλγόριθμο τους με την μοντελοποίηση και των δύο υποθέσεων σαν Gaussian 2<sup>ov</sup> βαθμού με μηδενική τιμή, στην περίπτωση αυτή έχω

$$L(x) = \frac{p(D(x) | H_1)}{p(D(x) | H_0)} = \frac{\sigma_0^N}{\sigma_1^N} \exp \left\{ - \sum_{y \in \Omega_x} D^2(y) \left( \frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_0^2} \right) \right\} \text{ (block version)}$$

Με τις παραμέτρους διασποράς να υπολογίζονται από περιοχές στην εικόνα διαφοράς  $D$ , για το  $s_0$  από τις περιοχές που δεν άλλαξαν ( $D(x)$  μικρό), για το  $s_1$  από τις περιοχές που έχω αλλαγή ( $D(x)$  μεγάλο).

Υπάρχουν και άλλες εργασίες με ανάλογο orientation πάνω στα  $H_i$ . Οι αλγόριθμοι που Gaussian modeling/ Poisson / Euler Number κτλ που χρησιμοποιήσαμε εμείς και αναλυτικά θα παρουσιαστούν παρακάτω στηρίζονται στην ανάλογη μεθοδολογία για την εύρεση του  $H_0$  όμως μόνο. Οι Bruzzone and Prieto [22] παρατήρησαν πως αν και ότι αν και οι διασπορές (όπως τις υπολογίσαμε παραπάνω) μπορούν να λειτουργήσουν σαν ένα καλό αρχικό estimation, οι χρήση μονό αυτών δεν δίνει και τα καλύτερα αποτελέσματα. Έτσι προτείνουν μια τεχνική αυτόματου change detection που ενοποιεί τα  $H_i$  σε μια και μόνη κατανομή

$$p(D(x)) = p(D(x) | H_0)P(H_0) + p(D(x) | H_1)P(H_1)$$

Οι μέσες τιμές και οι διασπορές των  $H$  υπολογίζονται τώρα με μια μέθοδος EM (expectation-maximization) [23] αρχικοποιούμενης με ανάλογο τρόπο με την μέθοδο Aach. Η μέθοδος τους όμως είναι πιο γενική, με τις  $D(x)$  να σαν συνδυασμός δυο non-parametric κατανομών που λαμβάνονται μέσω της διαδικασίας προσέγγισης Parzen [24] και βελτιστοποιούνται περιοδικά μέσω της EM.

## C) ΣΥΝΔΥΑΣΤΙΚΑ ΜΟΝΤΕΛΑ ΠΙΘΑΝΟΤΗΤΩΝ

Η κατηγορία έχει προταθεί από τους Black et al [25], που πρώτοι περιέγραψαν μια πρωτοποριακή προσέγγιση στον υπολογισμό των αλλαγών. Αντί λοιπόν της κατηγοριοποίησης των pixels στην  $D(x)$  σαν αλλαγή/ μη αλλαγή, κατηγοριοποιούνται με βάση της αιτίας που έχουν αλλάξει ή όχι. Έτσι, υπάρχουν εδώ (μεταξύ άλλων) οι classifiers 1) Αντικείμενο ή κίνηση της κάμερας 2) Φαινόμενα φωτεινότητας 3) Αντανακλάσεις (ξεχωριστά από 2) 4) «εικονικές» αλλαγές πχ αντικείμενο που μένει ακίνητο αλλά «χαλάει» η γεωμετρία του για κάποιο λόγο (όλες οι κατηγορίες ευρίσκονται με τη βοήθεια νευρωνικού που έχει γίνει training με ανάλογο dataset). Μια 5<sup>η</sup> κατηγορία κρατά τα pixel που δεν μπορεί να αντιστοιχίσει ο αλγόριθμος στα παραπάνω. Με χρήση του νευρωνικού λοιπόν και μέσω EM κάνει αυτή την κατηγοριοποίηση στο  $D(x)$ , λαμβάνοντας όμως υπόψη και τη πληροφορία στις εικόνες από τις οποίες προήλθε το  $D(x)$ . Ο αλγόριθμος αυτός είναι αρκετά ισχυρός και (πράγμα σπάνιο για τους change detection algorithms) μοναδικός στη μεθοδολογία του.

### 2.5.2 PREDICTIVE MODELS

Εδώ προσπαθούμε να υπολογίσουμε τους συσχετισμούς μεταξύ των γειτονικών pixels τόσο στο χώρο (spatial) όσο και στο χρόνο (temporal). Όπως θα φανεί και με τους δικούς μας αλγόριθμους οι κατηγοριοποιήσεις είναι σχετικές αφού όλοι σχεδόν οι αλγόριθμοι χρησιμοποιούν πιθανοτικά και spatiotemporal στοιχεία (με περίπου ίδια έννοια, non-adaptive προσέγγιση αρχικά στο ίδιο το pixel, adaptive σε σχέση με τα γειτονικά του έπειτα). Γίνεται όμως αυτή η αναφορά ώστε να κατανοήσουμε περισσότερο το state of the art των τεχνικών και να εισαχθούμε στην ανάλυση συγκεκριμένων αλγορίθμων παρακάτω.

## A) ΧΩΡΙΚΑ ΜΟΝΤΕΛΑ

Μια κλασσική change detection μεθοδολογία είναι να μοντελοποιήσουμε την ένταση κάθε pixel ως πολυώνυμο σε συνάρτηση των συντεταγμένων του. Με άλλα λόγια

$$\hat{I}_k(x, y) = \sum_{i=0}^p \sum_{j=0}^{p-i} b_{ij}^k x^i y^j$$

όπου  $p$  η τάξη του πολυώνυμου. Οι HSU et al [26] πρότειναν ένα γενικευμένο likelihood ratio test χρησιμοποιώντας σταθερό ή γραμμικά μεταβαλλόμενο μοντέλο γι' αυτή την αναπαράσταση σε κάθε block της εικόνας. Το  $H_0$  (δηλ. όταν δεν έχω πραγματική αλλαγή) στα ανάλογα block στις προς σύγκριση εικόνες, είναι η υπόθεση ότι οι συντελεστές του παραπάνω πολυωνύμου στα block αυτά **είναι ίδιοι**  $b_{ij}^0$ , ενώ αν  $H_1$  (έχω αλλαγή) τότε οι συντελεστές αυτοί είναι **διαφορετικοί**  $b_{ij}^1, b_{ij}^2$ .

Σε κάθε περίπτωση, οι παράμετροι  $b_{ij}^k$

υπολογίζονται μέσω least-square fit στις τιμές έντασης στα blocks της εικόνας. Το likelihood ratio υπολογίζεται μέσω

$$F(x) = \frac{\sigma_0^{2N}}{\sigma_1^N \sigma_2^N}$$

Όπου  $N$  ο αριθμός των pixels σε κάθε block,  $\sigma_1$  η διασπορά από την πολυωνυμική σχέση που αναπαριστά το block στη  $I_1$ ,  $\sigma_2$  η διασπορά από την πολυωνυμική σχέση που αναπαριστά το block στη  $I_2$ ,  $\sigma_0$  η διασπορά στην πολυωνυμική αναπαράσταση και των δύο blocks ταυτόχρονα. Το threshold στο γενικευμένο likelihood ratio test μπορεί να ευρεθεί μέσω ενός t-test (για σταθερό μοντέλο) ή ενός F-test (για γραμμικό μοντέλο ή quadratic μοντέλο). Οι HSU et al έκαναν χρήση και των τριών μοντέλων σε δοκιμές και συμπέραναν πως το quadratic model ήταν το καλύτερο σε επιδόσεις.

Μια επέκταση του παραπάνω αλγορίθμου έχει προταθεί από τους Skifstad and Jain [25]. Εδώ, προτείνεται μια στατιστική μετρική

$$T(\chi) = \sum_{y \in \Omega_\chi} \left( \frac{\partial \hat{I}_1}{\partial \chi}(y) - \frac{\partial \hat{I}_2}{\partial \chi}(y) + \frac{\partial \hat{I}_1}{\partial y}(y) - \frac{\partial \hat{I}_2}{\partial y}(y) \right)$$

Όπου, οι τιμές έντασης  $\hat{I}_j(x)$  μοντελοποιούνται σαν quadratic ( $a\chi^2 + b\chi + c$ ) συναρτήσεις των συντεταγμένων των pixels. Η μετρική αυτή συγκρίνεται με ένα εμπειρικό κατώφλι για να αποφασίσουμε τελικά το αλλαγή/ μη αλλαγή στο pixel. Αφού η μετρική περιλαμβάνει μόνο τοπικές παραγώγους, είναι ανεξάρτητη των γραμμικών αλλαγών στην ένταση. Όπως με το homomorphic filtering, υποθέτουμε πως οι εναλλαγές του φωτισμού λαμβάνουν χώρα στις χαμηλές χωρικές συχνότητες ενώ οι «πραγματικές» αλλαγές σε υψηλότερες. Επίσης υποθέτουμε ότι οι πραγματικές αλλαγές στα αντικείμενα θα αντικατοπτρίζονται από διαφορετικούς

συντελεστές στις quadratic συναρτήσεις που οι τοπικές παράγωγοι αυτών θα τους διατηρούν.

## B) ΧΡΟΝΙΚΑ (TEMPORAL) ΜΟΝΤΕΛΑ

Όταν οι αλλαγές προκύπτουν σε μια λογικά «συγκεκριμένη» σειρά εικόνων ( πχ ένας άνθρωπος που κινείται κτλ όχι άσχετες μεταξύ τους ), είναι εμφανές διαισθητικά ότι μπορούμε να κατευθύνουμε την έρευνα μας στη μελέτη των διαφορών μεταξύ των ίδιων pixels σε διαφορετικές χρονικές στιγμές.

Πολλοί μελετητές έχουν προτείνει μοντέλα των εντάσεων στα pixels κατά τις χρονικές μεταβολές αυτών ως μια «αυτοεξελισσόμενη» διαδικασία (autoregressive-AR). Μια από τις πρώτες παρόμοιες μελέτες είναι των Elfishawy et al [27]. Οι Jain και Chau [28] υπέθεσαν ότι κάθε pixel είναι όμοια και ανεξάρτητα κατανομημένο στην ίδια (χρονικά μεταβαλλόμενη) Γκαουσιανή κατανομή σε σχέση με το χρονικά προηγούμενο του. Με αυτή την υπόθεση, υπολόγισαν μέγιστες likelihood προσεγγίσεις της μέσης τιμής, της διασποράς και της συσχέτισης κάθε pixel στο χρόνο, χρησιμοποιώντας τα αποτελέσματα αυτά στα likelihood tests όπου η  $H_0$  υπόθεση (null hypothesis-nochange) είναι ότι τα pixel θα είναι εξαρτώμενα, ενώ η  $H_1$  (change/ alternate hypothesis) ότι τα pixel ( οι εντάσεις αυτών) θα προέρχονται από διαφορετικές κατανομές (άρα ανεξάρτητα).

Παρόμοια, ο background estimation αλγόριθμος του Toyama [8] δύναται να επεκταθεί σε καθαρά change detection. Στη βάση του χρησιμοποιεί ένα Wiener φίλτρο για να προβλέψει την ένταση σε ένα pixel βασιζόμενος σε ένα γραμμικό συνδυασμό των εντάσεων του ίδιου pixel στις k προηγούμενες χρονικές στιγμές ( στα k προηγούμενα frames). Στα pixel όπου το λάθος στη πρόβλεψη είναι πολύ μεγαλύτερο από το αναμενόμενο τα καταχωρούμε ως αλλαγή. Οι μεταβλητές πρόβλεψης ανανεώνονται σε κάθε frame ( θα πούμε κάποια πράγματα για τα φίλτρα πρόβλεψης όπως Wiener/ Kalman / Benedict-Borner κτλ στο κεφάλαιο του tracking, τώρα επειδή δεν τα χρησιμοποιήσαμε στο change detection απλώς αναφέρουμε τους ανάλογους αλγόριθμους στο πεδίο αυτό). Αυτό ο Toyama το χρησιμοποιεί για background estimation, αλλά αποτελεί καλή λογική και για temporal change detection. Ο αλγόριθμος προσπαθεί ακόμα να κατηγοριοποιήσει σωστά σε ένα σύνολο κάθε ομοιογενώς χρωματισμένο κινούμενο αντικείμενο υπολογίζοντας το ιστόγραμμα της γειτονιάς (4 ή 8 connected ) των pixels που κατηγοριοποιεί ως αλλαγμένα και προσθέτοντας στα σύνολα αυτά pixels που «ταιριάζουν» στο αντικείμενο με κριτήρια απόστασης και χρωματικής ομοιότητας.

Ο Carlotto [29] παρατηρεί πως τα γραμμικά μοντέλα δεν αποδίδουν καλά, και γι' αυτό προτείνει μια μη γραμμική μοντελοποίηση των σχέσεων μεταξύ δυο εικόνων  $I_i(x), I_j(x)$  της ίδιας video sequence κάτω από την  $H_0$  υπόθεση (δηλ. θα έχουν σχέση όπως υποθέσαμε στα προηγούμενα αλλά μη γραμμική). Η βέλτιστη μη γραμμική συνάρτηση γι' αυτό  $f_{ij}$  , η κατά συνθήκη αναμενόμενη τιμή της έντασης του pixel  $I_i(x)$  δεδομένης της τιμής του ίδιου pixel στο προηγούμενο χρόνο  $I_j(x)$  Για μια sequence N εικόνων υπάρχουν  $N(N-1)/2$  συνολικά παράγωγα εικόνων σφάλματος

$$\varepsilon_{ij}(x, y) = [I_i(x, y) - f_{ij}(I_j(x))] - [I_j(x, y) - f_{ji}(I_i(x))]$$

Αυτές οι εικόνες μετά υπόκεινται σε κάποιο threshold για να έχω τις τελικές binary change masks. Η μελέτη του Carlotto επικεντρώθηκε στην εύρεση συγκεκριμένων χρονικών patterns ( δηλ. εύρεση συγκεκριμένων συμπεριφορών στα παραγόμενα error images) με την ταυτοποίηση N change decisions στο πιξελ με το αναμενόμενο pattern εισόδου (δηλ. τι έχω σαν αποτέλεσμα, τι περιμένω από τη παραπάνω θεώρηση).

Ο Clifton [30] χρησιμοποιεί ένα adaptive νευρωνικό δίκτυο για την εύρεση μικρής κλίμακας αλλαγών σε μια sequence πολυφασματικών εικόνων. Αυτό μπορεί να ειπωθεί ως μια unsupervised μέθοδο για learning παραμέτρων ενός μη γραμμικού predictor, ανάλογου του Carlotto. Τα πιξελς στα οποία η πρόβλεψη αποκλίνει πολύ κατηγοριοποιούνται σαν changed. Ο σκοπός είναι να διακρίνουμε «μη αναμενόμενες» μεταβολές (πχ που υπόκεινται σε φωτισμό κτλ αλλά καθαρή κίνηση), αλλά αυτή η «ενοσιολογική» προσέγγιση δείχνει να μην λειτουργεί καλά, διότι η ουσιαστική παραμετροποίηση των αλλαγών με τέτοιο τρόπο είναι πρακτικά άπειρη (και γι' αυτό τα νευρωνικά δίκτυα δεν πολυχρησιμοποιούνται στο πεδίο, αλλά μόνο σε 2<sup>ο</sup> επίπεδο –adaptive μέρος- με καλά ορισμένα κριτήρια για αλλαγές και πάνω πάντα σε σχέση με το συγκεκριμένο sequence).

### C) SHADING ORIENTATION MODEL

Όπως είδαμε πριν, η δημιουργία illumination-invariant αλγορίθμου αποτελεί σημαντικό τμήμα του pre-processing (illumination models σαν το homomorphic filtering είναι αρκετά δημοφιλή) , υπάρχουν όμως και αρκετοί αλγόριθμοι που το όλον τους ( με την έννοια της κύριας αλγοριθμικής τους λογικής ) βασίζεται σε τέτοια μοντέλα και δεν τα χρησιμοποιούν μόνο optional για το restoration του workload.

Η βασική λογική αυτών των αλγορίθμων είναι η σύγκριση του ποσοστού αλλαγής δύο εντάσεων του ίδιου pixels σε διαφορετικές χρονικές στιγμές

$$R(x) = \frac{I_2(x)}{I_1(x)}$$

με ένα κατώφλι (ορισμένο εμπειρικά ή με κάποια άλλη μέθοδο, όπως πχ τις προηγούμενες).

Ας δούμε σε γενικές γραμμές πως μπορεί να γίνει αυτό. Έστω δύο shading models για τις δυο εικόνες προς σύγκριση  $I_1(x)$  και  $I_2(x)$ , τα  $I_1(x) = I_{i1}(x)I_{o1}(x)$  και  $I_2(x) = I_{i2}(x)I_{o2}(x)$  (όπως ορίσαμε στο homomorphic filtering δηλ. συνολική ένταση = ένταση φωτισμού επί ένταση ανάκλασης), άρα ο παραπάνω τύπος γίνεται

$$\frac{I_2(x)}{I_1(x)} = \frac{I_{i2}(x)I_{o2}(x)}{I_{i1}(x)I_{o1}(x)}$$

Στο παραπάνω πηλίκο μπορώ να πω ότι όταν περίπου ξεκινά η αλλαγή οι παράγοντες της ανάκλασης,  $I_{o1}(x) = I_{o2}(x)$  (αλλά και διαισθητικά, όπως θα φανεί από το homomorphic filtering στα test μας) είναι κοντά και οι παράγοντες φωτισμού  $I_{i1}$  και  $I_{i2}$  είναι αυτοί που ουσιαστικά μετρούν στο πηλίκο αυτό. Άρα έχω περίπου

$$\frac{I_2(x)}{I_1(x)} = \frac{I_{I_2}(x)}{I_{I_1}(x)}$$

Μπορούμε ακόμα να υποθέσουμε ότι για τα blocks της εικόνας  $\Omega_x$  (έστω  $n \times n$  γειτονιές) όπου ικανοποιείται η  $H_0$  υπόθεση (δηλ. δεν έχω αλλαγή) ο παράγοντας φωτισμού είναι ίδιος για όλα τα πιξελς της γειτονιάς και άρα το  $R(x)$  ίδιο. Αυτό μοντελοποιεί μια  $H_0$  υπόθεση η οποία υποθέτει γραμμική εξάρτηση μεταξύ των ανάλογων blocks στα  $I_1, I_2$  σε σχέση με το  $R$ .

Οι Skifstad και Jain [31] προτείνουν την ακόλουθη μέθοδο για την προσέγγιση της γραμμικής αυτής εξάρτησης μεταξύ δυο block που αναφέρονται στην ίδια θέση (spatial) σε διαφορετικούς χρόνους  $\hat{I}_1(x)$  και  $\hat{I}_2(x)$ ,

$$\text{Πρώτα δημιουργούμε το } \eta(x) = \frac{1}{N} \sum_{y \in \Omega_x} (R(x) - \mu_x)^2$$

$$\text{Όπου το } \mu_x \text{ δίνεται από τη σχέση } \mu_x = \frac{1}{N} \sum_{y \in \Omega_x} R(x)$$

Όταν η  $n(x)$  ξεπερνά κάποιο κατώφλι μπορούμε να θεωρήσουμε ότι έχουμε αλλαγή.

Ακόμα μια μέθοδο που στηρίζεται σε θεωρία shading (illumination invariant) model έχει προταθεί από τους Durucan/Ebrahimi [32].

Ξαναγυρνώντας στη μέθοδο των Skifstad et al, οι Aach, Mester et al [33] κρίνουν το  $n(x)$  τεστ ως ad hoc και όχι καλά ορισμένο. Προτείνουν λοιπόν ένα διαφορετικό τεστ βασισμένο στην ίδια λογική.

Κάτω από την  $H_0$  υπόθεση, κάθε εικόνα υποτίθεται πως έχει τη μορφή μιας συνάρτησης

$$\begin{aligned} \hat{I}_1(x) &= \hat{I}(x) + \delta_1 \\ \hat{I}_2(x) &= k\hat{I}(x) + \delta_2 \end{aligned}$$

Όπου,  $\hat{I}_2(x)$  και  $\hat{I}_1(x)$  error free εκδόσεις των εικόνων, δ εκφράζουν noise blocks που ακολουθούν  $2^{\text{ου}}$  βαθμού κανονικές κατανομές  $N(0, \sigma^2)$ . Η προβολή του  $\hat{I}_2(x)$  σε ένα σύστημα ορθογώνιο στο  $\hat{I}_1(x)$  δίνεται ως

$$\hat{O}(x) = \hat{I}_2(x) - \left( \frac{\hat{I}_1(x)^T \hat{I}_2(x)}{\|\hat{I}_1(x)\|^2} \right) \hat{I}_1(x)$$

Αυτό είναι 0 όταν μόνο τα  $\hat{I}_1(x)$ ,  $\hat{I}_2(x)$  είναι γραμμικά εξαρτώμενα. Ο συγγραφέας απέδειξε ότι η probability density function της ένωσης στο  $H_0$  είναι προσεγγιστικά μια κατανομή  $\chi^2$  με  $N-1$  βαθμούς ελευθερίας και χρησιμοποίησε αυτό το δεδομένο τόσο για το significance όσο και το likelihood test.

### 2.5.3 ΕΠΙΛΟΓΟΣ – ΖΗΤΗΜΑΤΑ ΑΠΟΔΟΣΗΣ

Τα significance/likelihood tests δεν είναι τόσο κατηγοριοποίηση των αλγορίθμων όσο θεωρία και λογική που οι περισσότεροι από αυτούς προσεγγίζουν για την υλοποίηση τους. Γενικά, όπως θα δούμε παρακάτω σε συγκεκριμένους αλγόριθμους,

δεν μπορούμε γενικά να έχουμε μια αυστηρή κατηγοριοποίηση, αλλά ο κάθε αλγόριθμος χρησιμοποιεί λίγο από την κάθε παραπάνω θεωρία. Αυτό που διακρίνει κάθε αλγόριθμο είναι που περισσότερο ρίχνει το βάρος του, αν δηλ. γι' αυτόν το predictive modeling κομμάτι του είναι το πιο σημαντικό ή (πχ Aach modeling) δίνει το βάρος αλλού, στο preprossecing με homomorphic filtering και στο adaptive likelihood test.

Παρεμβάλω εδώ και κάποια ζητήματα απόδοσης και σύγκρισης.

### 2.5.3.1 CHANGE MASK CONSISTENCY

Το αποτέλεσμα ενός change detection αλγόριθμου όπου εφαρμόζεται ανεξάρτητα σε κάθε pixel (non-adaptive) είναι γενικά θορυβώδες, με την change mask να περιέχει απομονωμένα changed pixels, τρύπες στα objects που οριοθετούν πραγματική αλλαγή, disturbances στα όρια των changed objects κτλ. Εφόσον οι αλλαγές σε πραγματικά video sequences συνήθως προκύπτουν από μεταβολή της εμφάνισης ή της κίνησης «κλειστών» (solid) αντικειμένων με κάποιο (πάνω- κάτω) σταθερό μέγεθος και επίσης «κλειστών» boundaries, οι change detection algorithms προσπαθούν το τελικό αποτέλεσμα τους να είναι ανάλογο των παραπάνω.

Η απλούστερη τεχνική είναι η εφαρμογή median filter ή άλλου ανάλογου μορφολογικού μετασχηματισμού (dilations/erosions κτλ στο τελικό αποτέλεσμα του αλγόριθμου ώστε να αφαιρεθούν απομονωμένα pixels και να έχω πιο smoothed regions. Αυτές οι τεχνικές δεν δίνουν και το καλύτερο αποτέλεσμα (και εμείς παρακάτω γενικά θα τις αποφύγουμε στο post-prossecing) γιατί ακριβώς, όπως είπαμε πριν αρκετές φορές, το πρόβλημα του change detection δεν είναι κάτι το trivial, σχεδόν κάθε εικόνα και περιβάλλον έχει τις δικές του απαιτήσεις, συνεπώς δεν μπορεί να «ομογενοποιηθεί» η διαδικασία με την εφαρμογή ενός τετριμμένου και μικρής κλίμακας (μια γειτονιά 3x3 τις περισσότερες φορές) μετασχηματισμού.

Οι περισσότερες προσπάθειες για επίτευξη του consistency των regions στη σημερινή βιβλιογραφία επικεντρώνονται στην εφαρμογή θεωρίας αλυσίδων Markov (Markov Random Fields – MRFs), θεωρία που βρίσκει πολλές εφαρμογές γενικά στο Computer Vision (ο αλγόριθμος Aach πχ στηρίζεται σε αυτή). Μια ευρέως χρησιμοποιούμενη τεχνική τέτοιου τύπου εδώ είναι η εφαρμογή ενός Bayesian Network με την a priori πιθανότητα μιας change mask να δίνεται από

$$P(B) = \frac{1}{Z} \exp^{-E(B)}$$

όπου Z σταθερά και E(B) είναι ένας παράγοντας «εντροπίας»/ ποιότητας που έχει χαμηλή τιμή όταν η region B είναι smoothed και μεγάλη όταν δεν είναι. Πχ, οι Aach et al επιλέγουν E(B) ανάλογο του αριθμού των changed pixels θεωρώντας B= 1 pixels, αυτό που ελέγχο και βρίσκω αριθμό changed pixels τις 4 ή 8- connected γειτονιάς του.

Ανάλογες προσεγγίσεις υπάρχουν, πχ Kasetkasen / Varshney [34] (simulated annealing και εύρεση της optimal change mask), Bruzzone / Prieto [22] (όμοια μέθοδος με Aach et al) κτλ.

### 2.5.3.2 ΚΡΙΤΗΡΙΑ ΕΠΙΔΟΣΕΩΝ CHANGE DETECTION

Η απόδοση των αλγόριθμων μπορεί να κριθεί με το πιο προφανή και αποτελεσματικό στη πράξη τρόπο, με το να κοιτάζουμε το αποτέλεσμα και να εξετάσουμε την ποιότητα του οπτικά σύμφωνα πάντα με τις απαιτήσεις της εφαρμογής που έχουμε. Αυτό, αν και είναι πολύ εμπειρικό, γενικά είναι το πιο αξιόπιστο.

Μια άλλη μέθοδος είναι η δημιουργία ενός λεγόμενου Flicker Animation, δηλ. video που αποτελείται από δυο μόνο βασικές εικόνες που εναλλάσσονται. Επειδή εδώ έχουμε από πριν γνώση της δομής των εικόνων αυτών και του τι αλλάζει μεταξύ τους μπορούμε εύκολα να δούμε πως συμπεριφέρεται ο αλγόριθμος και αν τα αποτελέσματα του είναι σταθερά σε κάθε εναλλαγή. Δοκιμές εδώ μπορούν να γίνουν και αλλάζοντας το ρυθμό (frame rate) μετάβασης (άρα περιμένω κενή change mask αν frame  $i$  ίδιο με το frame  $i+1$  κτλ).

Η δημιουργία ωστόσο ενός συγκεκριμένου και εφαρμόσιμου test για γενική χρήση με πραγματικά videos δεν είναι εύκολη υπόθεση, γιατί όπως προείπαμε, κάτι τέτοιο θα πρέπει να μπορεί επακριβώς να ορίζει τι ακριβώς δίνει ένας Άριστος Αλγόριθμος και αυτό σε συνάρτηση με τους τύπους των video που μπορώ να έχω ( άπειρες επιλογές), το τύπο των σφαλμάτων (άπειρες επιλογές), των αλλαγών (άπειρες επιλογές) κα.

Παράλληλα στα παραπάνω, ακόμα και η μερική διατύπωση τέτοιων συνόλων (πχ με βάση συγκεκριμένων τύπων video) είναι κάτι το υπερβολικά χρονοβόρο. Αντί αυτού, πολλές φορές ακολουθούμε τη τακτική του ορισμού Του Ιδανικού Human Observer. Δηλ. ορίζουμε ποιος είναι ο άνθρωπος παρατηρητής που με την εμπειρία του μπορεί να κρίνει την απόδοση της μεθόδου, πχ σε σύστημα ακτινογραφιών ένας ανάλογος γιατρός κτλ. Όμως και αυτό δεν είναι αρκετό, διότι ακόμα και με αυτή τη στρατηγική, η κριτικές του αλγορίθμου μπορεί να διαφέρουν από άνθρωπο σε άνθρωπο, ακόμα και μεταξύ των «ειδικών». Μπορούμε όμως, να θέσουμε ένα ικανοποιητικό όριο στις παρατηρήσεις αυτές (αν πχ το 99 % των γιατρών θεωρεί πως το change detection σύστημα για εγκεφαλογράφημα έδωσε τον όγκο σε όλες τις περιπτώσεις τότε μπορώ να θεωρήσω πως ο αλγόριθμος λειτουργεί καλάς).

Στις ( στη πράξη εξαιρετικά σπάνιες ) περιπτώσεις που η γενική αλήθεια , δηλ. τι ακριβώς είναι αλλαγή, μπορεί να οριστεί (και συνήθως για από πριν γνωστά ως προς τα changes κτλ videos) υπάρχουν μια σειρά μετρικές για τον έλεγχο της ποιότητας του change detection. Πχ [35] ο Rosin διατυπώνει τις εξής :

-Αν **TP**= ο αριθμός των Pixels της change mask που **σωστά** τέθηκαν ως change ( True Positives)

-Αν **FP**= ο αριθμός των pixels στην change mask που **λανθασμένα** τέθηκαν ως changes (False Positives/ False Alarms)

-Αν **TN**= ο αριθμός των pixels στην change mask που **σωστά** τέθηκαν unchanged (True Negatives)

-Αν **FN**= ο αριθμός των pixels στην change mask που **λανθασμένα** τέθηκαν unchanged (False Negatives/ Misses)

Τότε μπορούν να οριστούν οι εξής μετρικές

- The Percentage Correct Classification  $PCC = \frac{TP + TN}{TP + FP + TN + FN}$

-The Jaccard Coefficient  $JC = \frac{TP}{TP + FP + FN}$



-The Yule Coefficient  $YC = abs(\frac{TP}{TP + FP} + \frac{TN}{TN + FN} - 1)$

πχ, στη 1<sup>η</sup>, όσο πιο μεγάλο **PCC** τόσο καλύτερη απόδοση έχω.

Κάποιος έπειτα μπορεί να δημιουργήσει γραφήματα με τους παραπάνω classifiers ως συνάρτηση τις τιμές σε κάθε change mask εικόνα, και να αντιπαραβάλει πχ το γράφημα για κάθε αλγόριθμο που υλοποιεί κτλ για πιο οπτικά κατανοητό αποτέλεσμα.

Στη πράξη, μπορούν να μας χρησιμεύσουν και άλλες μετρικές που έχουν να κάνουν με την γεωμετρία αντικειμένων και τη θέση τους στο χώρο. Οι περισσότερες από αυτές, αν όχι όλες είναι οι κλασσικοί shape descriptors που εδώ μπορούν να εφαρμοστούν για υπολογισμό στατιστικών μεγεθών που αντικατοπτρίζουν τη ποιότητα του change detection. Τέτοιες είναι:

**A) Euler Number** και γενικά μετρικές που απεικονίζουν τον αριθμό των αντικειμένων σε μια εικόνα. Η μετρική αυτή όπως είπαμε βρίσκει τον αριθμό των αντικειμένων (ή πιο σωστά παρέχει μια προσέγγιση του αριθμού αυτών), και ο αλγόριθμος τους είναι ο εξής:

i) Υπολόγισε το  $X = \text{Αριθμός «Αντικειμένων»}$

όπου το  $X =$  ο αριθμός των patterns στην εικόνα της μορφής

$X =$  the number of occurrences of the 2x2 pattern:

0	0
0	1

ii) Υπολόγισε το  $V = \text{αριθμός των οπών στην εικόνα}$   
 $V =$  αριθμός των patterns στην εικόνα της μορφής

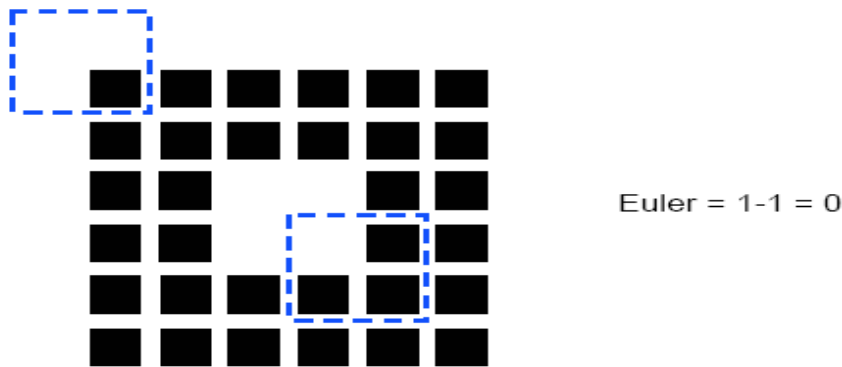
$V =$  the number of occurrences of the 2x2 pattern:

0	1
1	1

Τότε

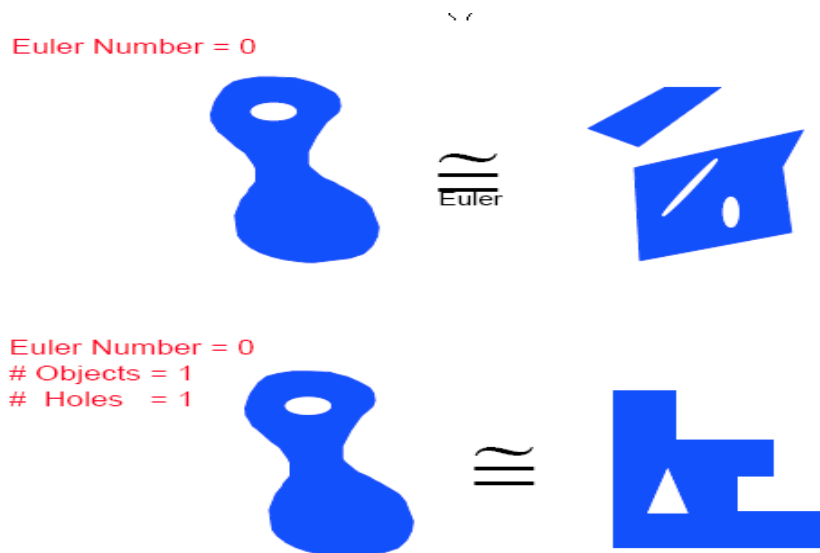
$$\text{Euler} = X - V$$

Πχ



### ΕΙΚΟΝΑ 6 ΠΑΡΑΔΕΙΓΜΑ ΥΠΟΛΟΓΙΣΜΟΥ EULER NUMBER (1)

Ο Euler number δίνει μόνο αριθμό αντικειμένων και δε μας λέει τίποτα για τη γεωμετρία κτλ αυτών πχ



The Euler Number is **not** a good shape descriptor.

### ΕΙΚΟΝΑ 7 EULER NUMBER (2)

Βλέπουμε, πως αν και τα αντικείμενα είναι πολύ διαφορετικά ο Euler number είναι ίδιος.

Ο Euler number παρέχει και τη δυνατότητα μιας ισχυρής μεθόδου change detection. Εδώ, αυτό που μπορούμε να πούμε, είναι πως όσο πιο μεγάλο αριθμό αντικειμένων βρούμε  $X$  σε συνδυασμό με μικρό Euler number στον έλεγχο των αποτελεσμάτων, τόσο μπορούμε να θεωρήσουμε πως ο αλγόριθμος δεν αποδίδει τόσο καλά, αφού αυτό δείχνει ότι έχω πολλά «σπαστά» αντικείμενα και κενά.

**B) Region Compactness**, αποτελεί κριτήριο για το πόσο «συγκεντρωμένη» είναι μια region ορίζεται ως

Compactness = (περίμετρος της region στο τετράγωνο) / Συνολική Region

Αυτό μπορεί να μεταφραστεί σε μια Binary region ως εξής

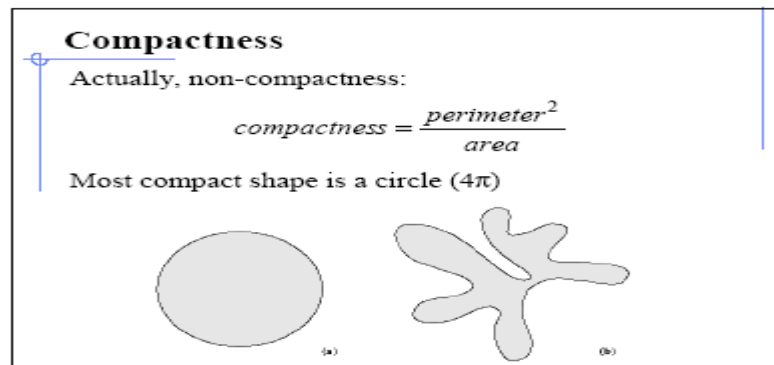
$$\text{Compactness} = K / N$$

Με  $K$ = τετράγωνο του αριθμού των pixels στην περίμετρο της region

$N$ = αριθμός των pixels της region

Όσο πιο μεγάλος τόσο η region είναι λιγότερο compact

πχ

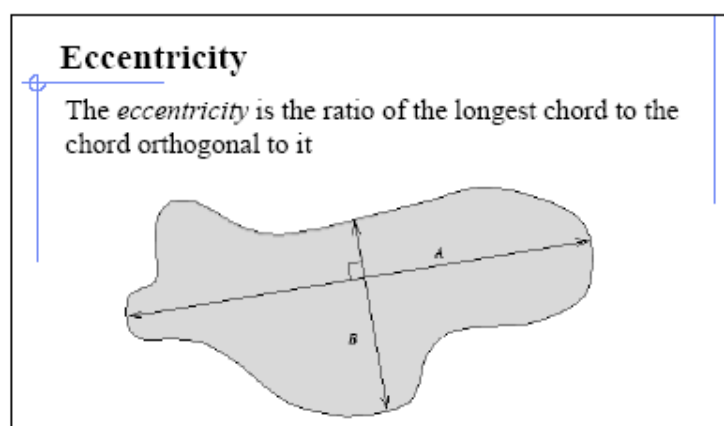


### ΕΙΚΟΝΑ 8 Region Compactness

Ο κύκλος α) είναι πιο compact από το β) ΑΝ ΚΑΙ ΚΑΤΑΛΑΜΒΑΝΟΥΝ ΤΟ ΙΔΙΟ ΠΕΡΙΠΟΥ ΕΜΒΑΔΟΝ

Αυτή η μετρική μπορεί να χρησιμοποιηθεί πχ αν κοιτώντας τη region παρατηρώ υψηλή τιμή τότε δεν έχω καλό αποτέλεσμα γιατί ο change detector δε δίνει ομοιογενείς περιοχές.

### C) Eccentricity



### ΕΙΚΟΝΑ 9

Δείγμα πάλι πόσο solid αντικείμενα έχω. Όσο πιο μεγάλο τόσο πιο μεγάλα αντικείμενα άρα πιο «συμπαγής» regions, δηλ. καλά αποτελέσματα στο change detection

**D) LABELING** αλγόριθμοι εν γένει. Αλγόριθμοι που τμηματοποιούν την εικόνα σε κλειστές περιοχές ( μαρκάροντας τις με κάποιο χρώμα ξεχωριστό από τις άλλες) , μας χρησιμεύει αφενός για έλεγχο ανάλογο με τον Euler Number (αριθμός περιοχών) αλλά και αφετέρου για την τμηματοποίηση των εικόνων που έχω ως change detection αποτέλεσμα για το επόμενο στάδιο ( Object Tracking). Υπάρχουν πολλοί Labeling αλγόριθμοι, πχ αυτός που χρησιμοποιήσαμε εμείς ο connected component labeling στην Binary εκδοχή του (0/1 εικόνα) ορίζεται ως:

1. Σκάνανε την εικόνα, πιξελ προς πιξελ, από το (0,0) πιξελ μέχρι το τελευταίο της (N, M)

2. Αν το pixel 1 τότε:

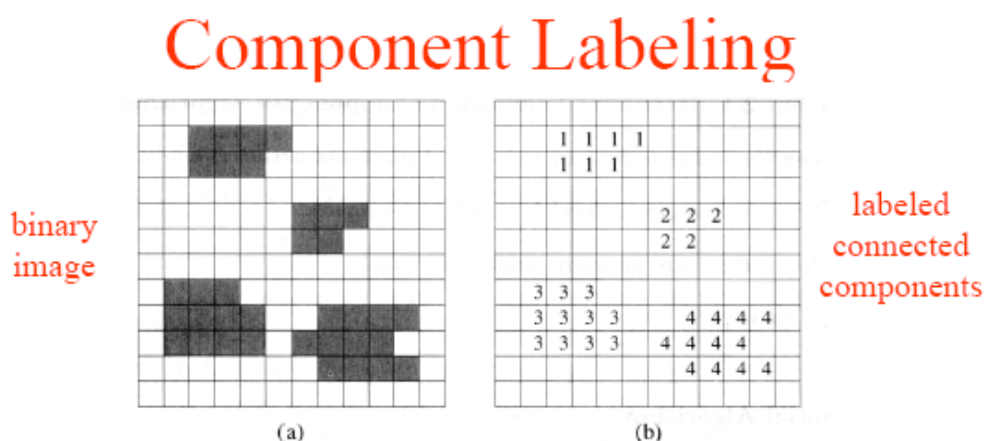
i) Αν τα γειτονικά pixels (πάνω αριστερά 8-connected neighbors ) είναι μηδέν τότε βάλε μια καινούρια ετικέτα στο πιξελ (πχ 5)

ii) Αν μόνο ένας πάνω αριστερά 8-connected neighbor είναι διάφορος του 0, τότε δώσε στο pixel την ετικέτα αυτού του γείτονα.

iii) Αν πάνω από ένας γείτονας του είναι διάφορος του 0, τότε δώσε στο pixel την ετικέτα κάποιου από αυτούς και θεώρησε όλες τις ετικέτες αυτές ισοδύναμες

iv) Αφού φθάσεις στο τέλος της εικόνας, ξανασκανάρησε τη και θέσε σε όλες τις ισοδύναμες τιμές μια μοναδική (πχ αν έχεις τα pixels με τιμή 5,3,14 ισοδύναμα θέσε σε όλα τιμή 5).

Όπου πάνω αριστερά 8-connected neighbor θεωρούμε τα εξής pixels, αν το προς εξέταση pixel είναι  $I(x,y)$  εννοούμε τα  $I(x-1,y-1)$ ,  $I(x-1,y)$ ,  $I(x,y-1)$  και  $I(x-1,y+1)$



**ΕΙΚΟΝΑ 10 ΑΠΟΤΕΛΕΣΜΑ LABELING ( a, η αρχική εικόνα και b, η labeled version) [12]**

Υπάρχουν αρκετές ανάλογες μετρικές, και πολλοί τρόποι να τις συνδυάσεις στον έλεγχο του αποτελέσματος. Κάποιους απ' τους παραπάνω χρησιμοποιήσαμε και εμείς.

## 2.6 ΕΠΙΛΟΓΗ ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΓΚΕΚΡΙΜΕΝΩΝ ΑΛΓΟΡΙΘΜΩΝ

Εδώ, ξεχωρίζουμε και ασχολούμαστε συγκεκριμένα με κάποιους αλγορίθμους. Στο προηγούμενο κομμάτι προσπαθήσαμε να κάνουμε μια εκτενή εισαγωγή ώστε ο αναγνώστης να καταλάβει περισσότερο ποιοτικά πως λειτουργεί το change detection. Για άλλη μια φορά επαναλαμβάνουμε ότι οι τεχνικές που εξετάσαμε αφορούν σταθερή κάμερα.

### 2.6.1 GAUSSIAN MODELING

Αν και η τεχνική αυτή αποτελεί κύριο θέμα πολλών papers και τη βάση πολλών μεταγενέστερων μεθόδων εν τούτοις παραμένει ισχυρότατο και διαχρονικό change detection εργαλείο. Η βασική της ιδέα είναι η μοντελοποίηση της συμπεριφοράς του θορύβου (και άρα η διατύπωση της null hypothesis  $H_0$ ) ως γκαουσιανής στο difference image των  $I_n$  και  $I_{n+1}$ . [5]

Υποθέτουμε πως ο θόρυβος ακολουθεί κανονική (GAUSSIAN) κατανομή με μέση τιμή 0,  $N(0, \sigma^2)$ . Τότε, το image differencing μπορεί να βρεθεί ως παίρνοντας την απόλυτη τιμή δίνει την κατανομή  $2N(0, 2\sigma^2)$  για θετικές τιμές.

Άλλοτε είναι χρησιμότερο να πάρουμε τις edge-detection εκδόσεις των αρχικών εικόνων αντί τις ίδιες, αφού τα edges είναι λιγότερο επηρεασμένα από θόρυβο. Τότε, θεωρούμε ότι ο θόρυβος στα edges μπορεί να προσεγγιστεί μέσω της κατανομής Rayleigh

$$R(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \quad x \geq 0$$

Αν η edge region της μορφής  $\sqrt{(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2}$ . Αναλύοντας την edge region ως δύο

ξεχωριστές τυχαίες μεταβλητές  $\mathbf{a}$  και  $\mathbf{b}$  θέλουμε να υπολογίσουμε την ένταση του θορύβου στην difference image  $\mathbf{d} = \mathbf{abs}(\mathbf{a} - \mathbf{b})$ .

Αρχικά, θεωρούμε τη συμμετρική σχέση  $\mathbf{c} = \mathbf{a} - \mathbf{b}$ . Η συνάρτηση της έντασης εδώ  $f_c(c)$

$$f_c(c) = \int_{-\infty}^{\infty} f_a(c-x) f_b(-x) dx$$

όπου  $f_a(c) = f_b(c) = R(x)$

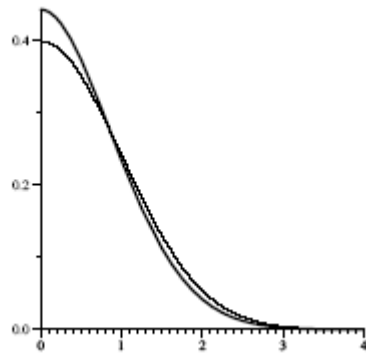
και με υπολογισμούς :

$$f_c(c) = \frac{\sqrt{\pi}}{8\sigma^2 e^{\frac{\pi^2}{4\sigma^2}}} \left[ 2\sigma - \frac{\chi^2}{\sigma} + 2\sqrt{\pi} \chi L_{-0.5}^{0.5}\left(\frac{-\chi^2}{4\sigma^2}\right) - \sqrt{\pi} \chi L_{0.5}^{0.5}\left(\frac{-\chi^2}{4\sigma^2}\right) \right]$$

όπου  $L_a^n(x)$  = generalised Laguerre πολυώνυμο

Ακόμα ισχύει  $d=abs(c)$ , άρα  $f_d(c)=2 f_c(c)$ , για  $\chi \geq 0$

Το γράφημα της  $f_c(c)$



σε αντιπαράσταση με αυτό σε μια  $N(0,1)$ .

Βλέπουμε ότι συμπίπτουν αρκετά, ιδιαίτερα στις άκρες. Άρα, μπορούμε να προσεγγίσουμε την  $f_d$  μέσω μιας κανονικής κατανομής. Αν επιλέξουμε κάποια thresholding τιμή  $\chi$ , η πιθανότητα λάθος κατηγοριοποίησης pixel

$$P_F = \text{erfc}\left(\frac{x}{\sqrt{2}\sigma}\right)$$

Συνεπώς, μπορούμε να επιλέξουμε ένα κατώφλι ώστε  $P_F$  μικρό (πχ 1%-5%).

Η παραπάνω μελέτη λειτουργεί απλώς ως παρουσίαση του τελικού τύπου του Gaussian Modeling, ακριβώς παραπάνω. Άρα, αν υποθέσω γκαουσιανή κατανομή του θορύβου και ότι έχω κάποιο τρόπο να βρω τη διασπορά του  $\sigma$ , τότε θέτοντας ένα ποσοστό σφάλματος, το ανεκτό για μας % που ο αλγόριθμος μας μπορεί να ταξινομεί θόρυβο ως πραγματικές αλλαγές, πχ 3% βρίσκω από το τελευταίο τύπο ποιο threshold  $x = T$  να επιλέξω.

Απομένουν δύο ερωτήματα να λυθούν:

- 1) Πως μπορώ να προσεγγίσω τη διασπορά  $\sigma$  του θορύβου στην εικόνα;
- 2) Ο υπολογισμός της  $\text{erfc}$  συνάρτησης ;

Και τα δύο, αποτελούν ζητήματα που αναλύουμε στο appendix A ( όπου συζητάμε διάφορες προσεγγίσεις για noise estimation και καταλήγουμε σε μια μέθοδο αρκετά δημοφιλή για change detection) και στο appendix B (η  $\text{erfc}$  αναλύεται μαθηματικά και εξηγούμε την υλοποίηση της σε κώδικα).

Συνεπώς ο αλγόριθμος της Gaussian modeling έχει ως εξής

1)Βρες το  $\sigma$  του θορύβου στο difference image

- II) Θέσε ένα ανεκτό σφάλμα στην πρόβλεψη πχ ο αλγόριθμος θα κατατάσσει πραγματικές αλλαγές ως θόρυβο με ποσοστό λιγότερο του 5%.
- III) Τότε το threshold θα πρέπει να επιλεγεί από τη σχέση

$$P_F = \operatorname{erfc}\left(\frac{x}{\sqrt{2}\sigma}\right)$$

Τα αποτελέσματα του γενικά πάντα είναι καλά [5] αλλά επίσης όπως θα δούμε στην δική μας εργασία, είναι πολύ γρήγορος, διότι εφαρμόζεται με ένα και μόνο scanning στην εικόνα, αλλά μπορεί να εφαρμοστεί και τμηματικά σ' αυτήν.

### 2.6.2 POISSON MODELING

Άλλες πάλι φορές μπορούμε να υποθέσουμε πως ο θόρυβος ακολουθεί κατανομή poisson. Τότε προκύπτει [5,11] ένας spatial modeling αλγόριθμος

-Τμηματοποιούμε πρώτα το difference image σε NxN παράθυρα

- Για κάθε τιμή threshold τώρα

i) Βρες σε κάθε παράθυρο τα pixels που η ένταση τους ξεπερνά το threshold. Αθροίζοντας των αριθμό αυτών των pixels σε όλα τα παράθυρα και διαιρώντας με τον αριθμό των παραθύρων προκύπτει μια μέση τιμή  $\bar{x}$ .

ii) Με ανάλογο τρόπο βρίσκουμε και τη διασπορά του αριθμού των pixels στην εικόνα πάνω από το threshold.

iii) Τότε, για το threshold που έχω επιλέξει προκύπτει η σχετική διασπορά

$$V_r = \frac{s^2}{\bar{x}}$$

Δηλ. διασπορά προς μέση τιμή των pixels που ξεπερνούν το threshold στην εικόνα

- Επέλεξε το threshold που μεγιστοποιεί τη σχετική διασπορά.

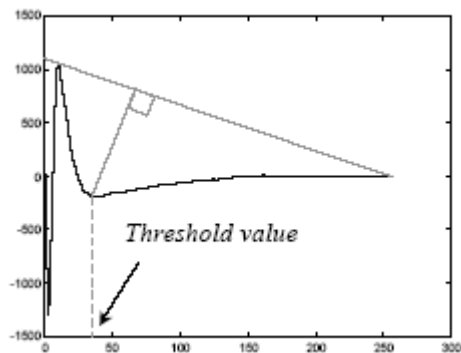
### 2.6.3 EULER NUMBER

Αντί της απαρίθμησης των changed regions, μπορούμε να επιλέξουμε την εύρεση του αριθμού Euler (όπως τον ορίσαμε πιο πριν) της εικόνας για κάθε ένα κατώφλι στην thresholded εικόνα του image difference.

Μια μη μεταβαλλόμενη περιοχή στην τιμή Euler για ένα εύρος κατωφλίων αντιστοιχεί σε μια «επίπεδη», ευθεία (plateau) περιοχή στο γράφημα των Euler Number σε συνάρτηση με τα κατώφλια, δηλ εκεί τα αντικείμενα στην εικόνα δεν μεταβάλλονται ως προς τον αριθμό τους και τη γεωμετρία τους στο περίπου. Συνεπώς, εκεί που ξεκινά η περιοχή αυτή, επιλέγω το κατώφλι.

Δυστυχώς, λόγω του θορύβου κτλ αυτό δεν ισχύει ακριβώς, η περιοχή «ηρεμίας» δεν είναι ακριβώς ευθεία, έχει μεταβολές, αν και μικρότερες από το υπόλοιπο γράφημα.

Μια λογική που μπορεί να ακολουθηθεί για να ξεπεραστεί το πρόβλημα, είναι η μοντελοποίηση του γραφήματος με ένα «φθίνων» εκθετικό e. Δηλ. με μια μορφή



**ΕΙΚΟΝΑ 11 ΤΟ ΓΡΑΦΗΜΑ ΤΟΥ EULER NUMBER ΟΠΩΣ ΠΡΟΚΥΠΤΕΙ ΣΕ ΣΥΝΑΡΤΗΣΗ ΜΕ ΤΟ THRESHOLD**

Και ένα κατάλληλο σημείο επιλογής του κατωφλίου ( και άρα, το κατώφλι που χωρίζει θόρυβο από περιοχές πραγματικής μεταβολής ) είναι η «γωνία» του γραφήματος, που βρίσκεται ως το σημείο με την μεγαλύτερη απόσταση από την ευθεία που ενώνει το max με το τέλος (τιμή κατωφλίου 255, 1 για κανονικοποιημένη εικόνα, κτλ). Για να αποφευχθούν συγχύσεις λόγω πχ πολλών peaks αντί αυτής της μορφής κτλ, **πάντα η ευθεία δημιουργείται από την πρώτη θετική peak στο γράφημα και την τιμή στο άνω όριο της έντασης.**

Δηλ. ανακεφαλαιώνοντας ο αλγόριθμος

- Για κάθε threshold [0...255] στο difference image βρες Euler Number

- i) Βρίσκω το μέγιστο euler number, έστω για κατώφλι T

ii) Βρίσκω την ευθεία που προκύπτει από το σημείο του μέγιστου euler number και του Euler(255)

Θα είναι μια ευθεία  $f = ax + b$

Με

$Euler(255) = 255a + b$

$Euler(T) = Ta + b$

- Τώρα για να βρω για ποιο T στη συνάρτηση του Euler Number έχει μέγιστη απόσταση από την ευθεία αυτή κάνω το εξής:

Για κάθε σημείο [0..255] της συνάρτησης Euler(T) βρίσκω μια ευθεία την

$Y = -ax + b1$ , θα είναι δηλ. κάθετη στην  $y = ax + b$

Συνεπώς για κάθε T:

βρίσκω το b1 του, από εκεί βρίσκω το κοινό σημείο της Y με την f  
βρίσκω απόσταση του σημείου αυτού με το Euler(T)

Κρατώ το T και κάνω τελικά thresholding στο difference image για το οποίο η απόσταση αυτή μέγιστη.



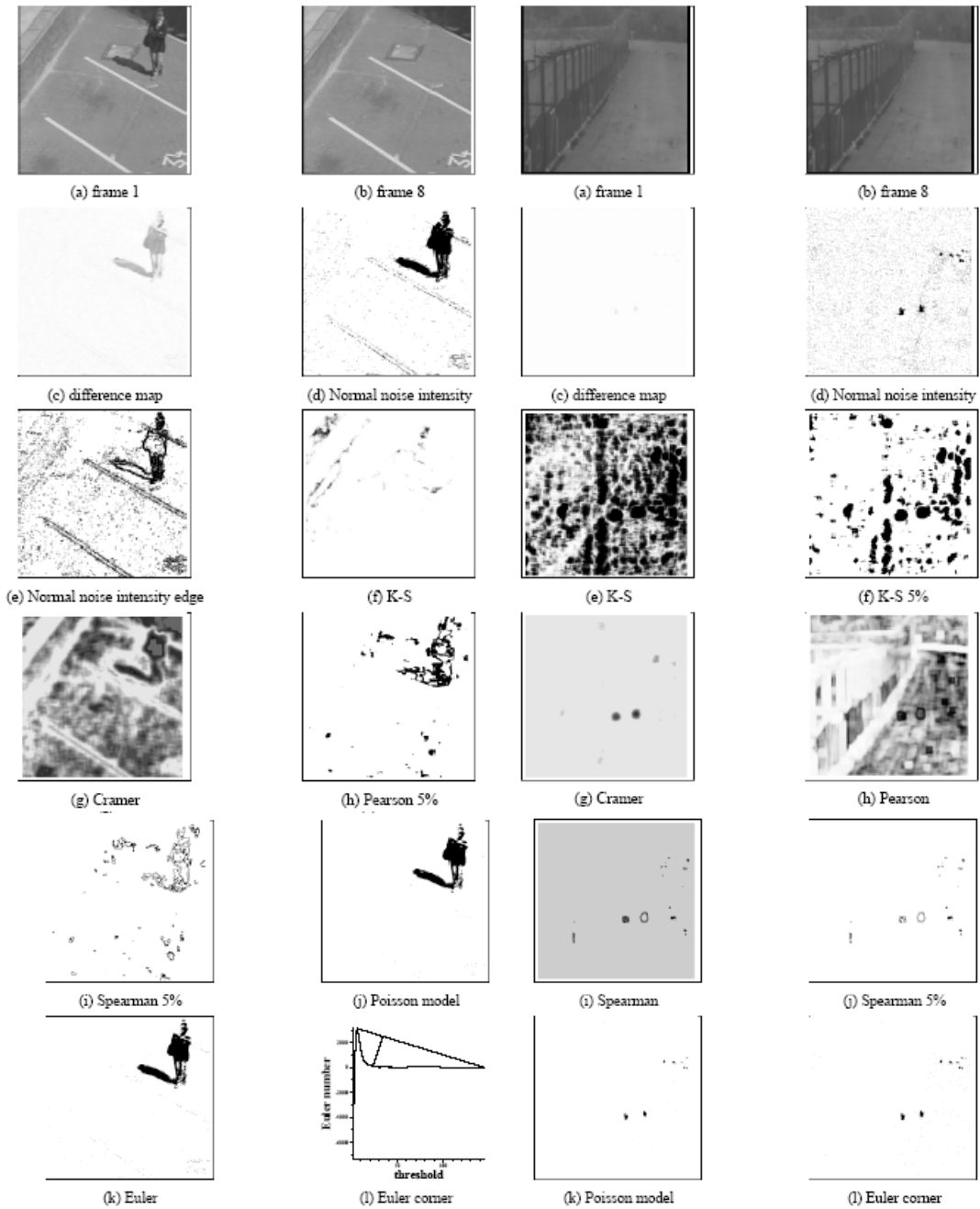


Figure 2: Pedestrian crossing car park

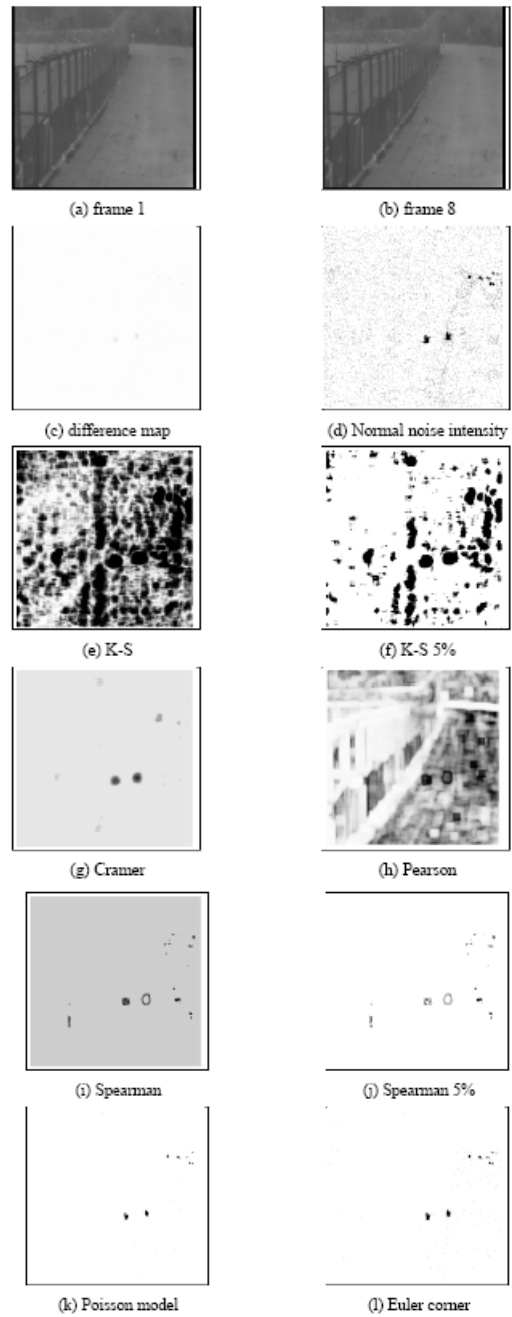


Figure 3: srdb018 image sequence containing moving rabbits

## ΕΙΚΟΝΑ 12 [5] TESTS ΤΟΥ ROSIN. GAUSSIAN/POISSON/EULER ΓΕΝΙΚΑ ΜΕ ΚΑΛΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

### 2.6.4 ΜΕΘΟΔΟΣ AACH

Ο αλγόριθμος αυτός [13,14,15] συγκεντρώνει μια σειρά στοιχεία που κάνουν τη μελέτη του αρκετά ενδιαφέρουσα. Στη βάση του, 1<sup>ο</sup> μέρος, ακολουθεί μια λογική παρόμοια με το Gaussian Modeling όπως το είδαμε παραπάνω, μια spatiotemporal adaptive βελτίωση (2<sup>ο</sup> μέρος, όχι κάτι σημαντικό, μια πρακτική περισσότερο

προσέγγιση) αλλά κυρίως με την «επιμονή» στο homomorphic preprossecing μας βάζει σε ένα ερώτημα για τη σημασία του preprossecing.

## A. Context-adaptive Motion Detection

Ο προορισμός του motion detection είναι η δημιουργία μιας μάσκας  $q$  αποτελούμενοι από binary ετικέτες  $q(k)$  κάθε pixel για την προς έλεγχο εικόνα. Οι ετικέτες αυτές, είναι είτε «1» (το pixel είναι διαφορετικό σε σχέση με την εικόνα αναφοράς), είτε «0» αν είναι ίδιο με το ανάλογο στην εικόνα αναφοράς ( background image). Για το προσδιορισμό των ετικετών αυτών  $q(k=i)$  για το pixel  $i$ , ξεκινάμε με την (σε ασπρόμαυρο) διαφορά (εικόνα αναφοράς- εικόνα προς σύγκριση) , και συγκρίνουμε το άθροισμα των απόλυτων διαφορών  $\Delta_i$  ΣΕ ΚΑΘΕ ΠΑΡΑΘΥΡΟ  $w_i$  αποτελούμενο από  $N$  pixels με κέντρο το pixel  $i$  με ένα κατώφλι  $T$  :

$$\Delta_i = \frac{2\sqrt{2}}{\sigma_u} \sum_{k \in w_i} |d(k)| \quad , \text{ if } \Delta_i > T \text{ then pixel } 1$$

$$\text{else} \quad 0$$

όπου  $\sigma_u$ , η διασπορά του (gaussian) θορύβου των διαφορών στο difference image. Η διασπορά αυτή, μπορεί εύκολα να προσεγγιστεί ( appendix A). Το  $\Delta_i$  , ακολουθεί  $\chi^2$  κατανομή με  $2N$  βαθμούς ελευθερίας. Το  $T$ , μπορεί από εδώ να προσδιοριστεί, ως  $a = p(\Delta_i > T | H_0)$  . Αυτή η διαδικασία καλείται significance test, και ισούται με την πιθανότητα να είναι το  $\Delta_i > T$  για την υπόθεση  $H_0$  ότι το pixel  $d(k)$  ανήκει σε θόρυβο. Για ένα δοθέν  $a$  (λάθος επαλήθευση ότι το pixel είναι κανονική κίνηση, την «ευαισθησία»  $a$  αυτή, την ορίζουμε εμείς ) , το  $T$  δύναται να βρεθεί από τους πίνακες της  $\chi^2$  κατανομής.

Η παρακάτω διαδικασία δε δίνει τέλεια αποτελέσματα και μπορεί να παράγει τόσο λάθος «0», όσο και λάθος «1». Βελτίωση του αποτελέσματος μπορεί να επιτευχθεί με την μείωση του κατωφλίου στις «αλλαγμένες» περιοχές (πχ περιοχές με πολλές αλλαγές κτλ) και τη αύξηση έξω από αυτές. Αυτό απαιτεί γνώση των changed region και του background, που την παίρνουμε από το προηγούμενο βήμα και μοντελοποιούμε κατάλληλα τις νέες change masks μέσω Gibbs/Markov τυχαία πεδία

Σκανάρουμε λοιπόν την εικόνα που παρήχθη από την 1<sup>η</sup> διαδικασία ( binary 0/1 ) από πάνω αριστερά μέχρι κάτω, δεξιά, pixel προς pixel. Για κάθε pixel κοιτάμε την 3x3 (8-neighbor) του και μετράμε τον αριθμό των «changed» pixels που προήλθαν από την προηγούμενη μέθοδο. Στη πράξη, μόνο τα 4-neighbor είναι άμεσα διαθέσιμα, τα άλλα τα έχουμε κρατώντας την change mask από τα προηγούμενα . Βρίσκουμε λοιπόν μια τιμή, ένα νέο κατώφλι και το εφαρμόζουμε στη γειτονιά

$$t(n_i) = T + (4 - n_i)B$$

όπου  $0 \leq n_i \leq 8$  , τα pixels της γειτονιάς, με  $B > 0$  και user defined

Καθαρά φαίνεται, πως το  $t(n_i)$  μειώνει , τρόπο τινά , τα αποκομμένα μικρά regions (errors) . Αυτό για κάθε pixel, μεταφέροντας τ' αποτελέσματα από pixel προς pixel κτλ.

## B. Spatio-temporal Noise Detection ( Βελτίωση του παραπάνω ώστε να «προσεγγίσω τη 3-Δ πληροφορία από την 2-Δ εικόνα»)

Ως τώρα, ο αλγόριθμος παίρνει υπόψη τη 2-D spatial πληροφορία , επιχειρούμε μια μετατροπή του ώστε να παίρνει υπόψη και την temporal. Όταν επεξεργαζόμαστε πραγματικά 3-Δ datasets , πχ εικόνες τομογράφου, οι 3-Δ γείτονες κάθε pixel μπορούν να υπολογιστούν. Σε temporal image sequences , μόνο αιτιατή πληροφορία είναι γνωστή. Αντιμετωπίζουμε ένα boundary πρόβλημα όπου το corresponding pixel του προηγούμενου frame για κάθε προς έλεγχο frame, στην ίδια θέση, υπολογίζεται στη 3-Δ γειτονιά. Πάλι, βρήκαμε την εξής λύση

$$t(n_i) = T + (4 - n_i)B + C\left(\frac{1}{2} - n_c\right)$$

όπου B,C user defined, και θετικά

$n_c=1$ , if pixel of the frame marked as changed

$n_c=0$ , if pixel of the frame is marked as unchanged

## C. Homomorphic Pre-filtering

Ο Aach δίνει ιδιαίτερη σημασία στο στάδιο του homomorphic prefiltering, αν και αποτελεί τμήμα του preprossecing.

Η φωτεινότητα σε μια εικόνα (intensity) δημιουργείται από μια προσπίπτουσα ακτινοβολία (illumination), η οποία αντανakλάτε στις επιφάνειες των αντικειμένων στη σκηνή. Σαν πρώτη προσέγγιση , για Lambertian επιφάνειες αντικειμένων , η ένταση στο  $\tau$  frame μιας ακολουθίας εικόνων δίνεται ως

$$y_\tau(k) = i_\tau(k)r_\tau(k)$$

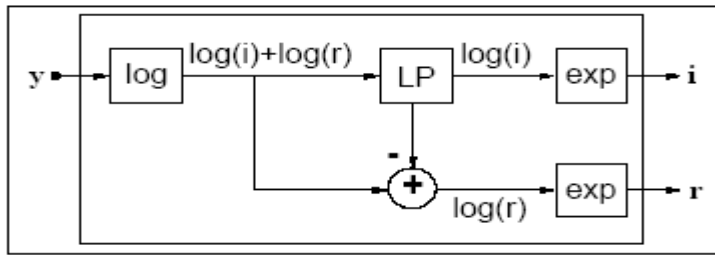
δηλ. με απλά λόγια σε κάθε frame, η ένταση κάθε pixel είναι το γινόμενο της φωτεινότητας  $i$  (illumination) και της αντανάκλασης ( reflectance )  $r$ . Αφού η  $r$  περιέχει **κυρίως** πληροφορία για το αντικείμενο αυτό καθ' αυτό (ενώ η  $i$  μπορεί να είναι πχ κάποια λάμπα φωτίζει τη σκηνή) , επιχειρούμε να απομονώσουμε  $i$  και  $r$ , και να χρησιμοποιήσουμε **μόνο** το δεύτερο για change detection με τον παραπάνω, όπως τον αναπτύξαμε, αλγόριθμο. Οι Oppenheim et Al [16], βρήκαν ότι οι εικόνες έχουν βασικά δύο **φασματικούς συντελεστές**, μια «κορυφή» στις χαμηλές συχνότητες, που σχετίζεται **κυρίως** με το illumination, και ένα μέσης και υψηλής συχνότητας «πεδιάδα» (plateau) που σχετίζεται **κυρίως** με τη reflectance.

Έτσι, μπορούμε να απομονώσουμε τις δύο μεταβλητές, με την εφαρμογή λογάριθμου και μετά ενός **γραμμικού low pass Filter** (περισσότερα **Appendix C, εμείς στην εργασία το υλοποιήσαμε με 3x3 averaging** ).

$$\log(y_\tau(k)) = \log(i_\tau(k)) + \log(r_\tau(k))$$

βέβαια, ο λογάριθμος είναι μη-γραμμική συνάρτηση και συνεπώς λογικά επηρεάζει τα  $i$  ,  $r$  αλλά [16] **πρακτικά αμελητέα**. Η παραπάνω σχέση ισχύει ακόμα και σε περιπτώσεις όπως camera nonlinearities (gamma correction) , καθώς με  $\log(\text{gamma exponent}) = \text{simple gain factor}$  .

Έτσι, πριν την εφαρμογή του κυρίως detection αλγορίθμου έχω το εξής φιλτράρισμα, στις υποψήφιες εικόνες :



**ΕΙΚΟΝΑ 13 HOMOMORPHIC FILTERING**

Για το LP ( low pass filter ), η συγγραφική ομάδα δοκίμασε πολλά είδη ( binomial, Gaussian and rect ). Όλα τα φίλτρα έχουν το μειονέκτημα ότι όσο μειώνω το illumination τόσο χάνω και από το reflectance (χάνεται πληροφορία κτλ )». Μετά με ύψωση με exponential μπορούμε να πάρουμε τις μεταβλητές που θέλουμε. Μπορούμε να παραλείψουμε, για το σκοπό που δουλεύουμε, το exp στάδιο , αλλά τότε η προσέγγιση της διασποράς του θορύβου της κάμερας  $\sigma_u$  , γιατί ο log την κάνει εξαρτώμενη του σήματος , για την αρχική μας εξίσωση

$$\Delta_i = \frac{2\sqrt{2}}{\sigma_u} \sum_{k \in w_i} |d(k)| , \text{ if } \Delta_i > T \text{ then pixel } 1 \text{ else } 0 \quad (1)$$

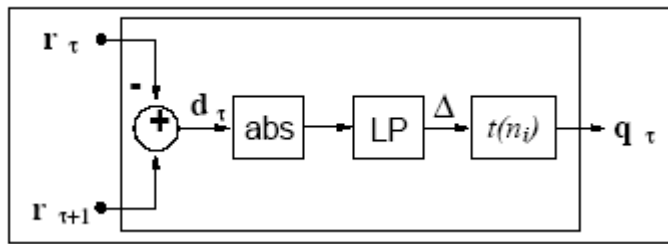


Η αρχική εικόνα έχει παρθεί από μια διαδοχική σειρά εικόνων, με δύο κινούμενα «τρενάκια». Κάθε εικόνα είναι διαστάσεων 320x240. Επιπρόσθετα, ένα φως διαπερνά «γρήγορα» τη σκηνή από αριστερά προς τα δεξιά. Στο συγκεκριμένο καρέ, βρίσκεται στο κέντρο της εικόνας. Η μεσαία εικόνα δείχνει το reflectance και η κάτω το illuminance κομμάτι της εικόνας όπως πάρθηκε από την εφαρμογή του Homomorphic filtering, LP ένα Gaussian low-pass με μέγεθος 31x31 pixels και κανονική διασπορά 14 να χρησιμοποιείτε για το όλο homomorphic. Το αποτέλεσμα γενικά καλό, όπως δείχνει και η εικόνα, αν και υπάρχει μια «λευκή περιοχή» (σαν «παχύ edge») όπου έχω blurring του αντικειμένου. Φυσικά, η «απογύμνωση» του reflectance από το illuminance δεν είναι τέλεια, αλλά για τους σκοπούς ενός motion detection αρκετά καλή.

**ΕΙΚΟΝΑ 14 HOMOMORPHIC FILTERING**

**Τελικό Σύστημα**

Έτσι αν, πχ  $y_t$  και  $y_{t+1}$  οι δύο προς σύγκριση εικόνες, πρώτα μέσω homomorphic απομονώνουμε τα reflectance τμήματα τους και τα περνάμε από το παρακάτω :



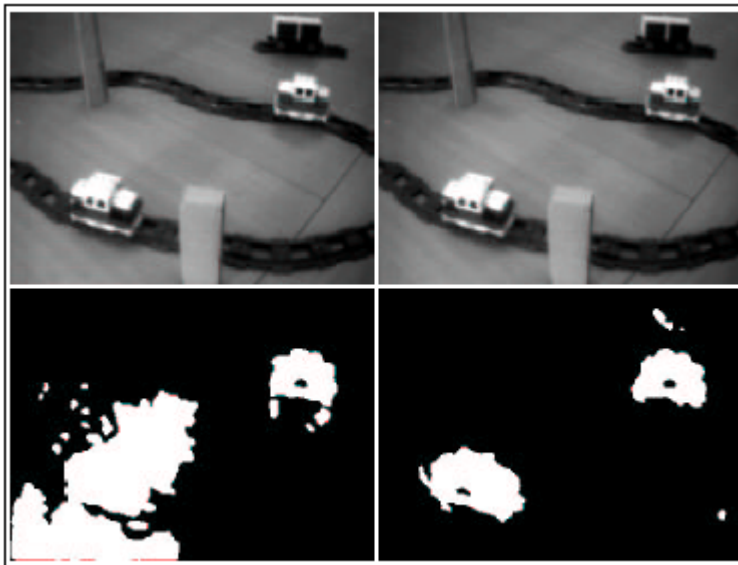
**ΕΙΚΟΝΑ 15 Ο ΣΥΝΟΛΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ**

όπου LP εδώ, ένα moving-average φίλτρο (5x5) για τον υπολογισμό του αθροίσματος

$$\sum_{k \in w_i} |d(k)|, \text{ στην αρχική εξίσωση (1) για τον υπολογισμό του } \Delta_i$$

### Δοκιμές και επισημάνσεις

Έστω 2 καρέ από το σετ με τα «τρενάκια» που είπαμε παραπάνω, με το φωτισμό να διαπερνά τη φωτογραφία κτλ

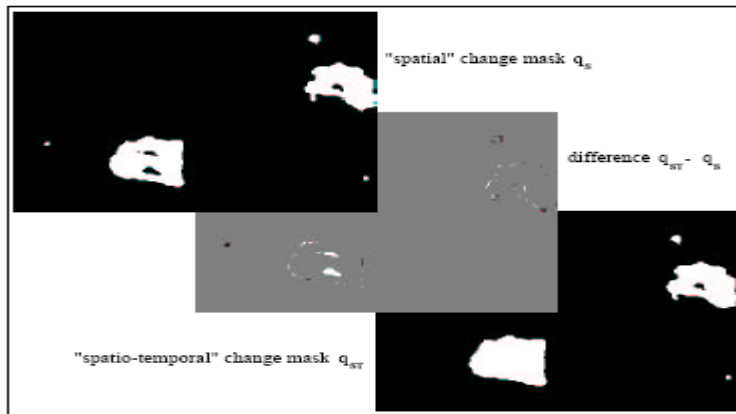


**ΕΙΚΟΝΑ 16 ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ ΜΕΘΟΔΟΥ ΑΑΧΗ ΠΡΩΤΑ ΧΩΡΙΣ ΚΑΙ ΜΕΤΑ ΜΕ HOMOMORPHIC**

Έχοντας προηγουμένως φιλτράρει τις εικόνες με homomorphic βλέπουμε (κάτω δεξιά) το change detection καλύτερο από χωρίς πρώτα φιλτράρισμα (κάτω αριστερά)

Οι δοκιμές έγιναν με  $\alpha = 0.0005$ ,  $B=15$ ,  $C=8$ . Το  $\sigma_u^2$  προσεγγίστηκε 5.

Ας δούμε ακόμα και το παρακάτω



**ΕΙΚΟΝΑ 17**

παρατηρούμε ότι η spatio-temporal ( $C > 0$ ) βελτίωση οδηγεί σε περαιτέρω «κλείσιμο» των changed περιοχών και «σβήσιμο» μικρών αποκομμένων περιοχών ενώ το homomorphic filter βελτιώνει σημαντικά τον αλγόριθμο.

Όπως είδαμε λοιπόν, ο Aach δίνει βαρύνουσα σημασία στην προεπεξεργασία των εικόνων και κυρίως στο homomorphic filtering. Δείχνει πως η υλοποίηση του και εφαρμογή βελτιώνει σημαντικά το change detection αλγόριθμο. Για αυτό και εμείς υλοποιούμε homomorphic στα δικά μας tests.

## **ΚΕΦΑΛΑΙΟ 3: ΑΛΓΟΡΙΘΜΟΙ OBJECT TRACKING**

- Εννοιολογικός Προσδιορισμός, Απαιτήσεις και Προβλήματα
- Παρουσίαση Του Γενικευμένου Μοντέλου Σταθερής Ταχύτητας
- Παρουσίαση της οικογένειας των a-b (a-b-k) Filters
- Linear Kalman Filter, Στοιχεία Υπολογισμού Αυτού, Πλεονεκτήματα
- Εισαγωγή Στην Έννοια Του non-linear Filtering και του non-linear Kalman Filter (EKF)

### 3. ΑΛΓΟΡΙΘΜΟΙ OBJECT TRACKING

#### 3.1 ΕΙΣΑΓΩΓΗ

Με το object tracking επιχειρούμε τη παρακολούθηση της κίνησης του αντικειμένου στο χρόνο. Κατά βάση προσπαθούμε να προβλέψουμε «επακριβώς» τη θέση τους σε κάθε frame  $n$ , τη ταχύτητα τους, ενίοτε όμως και την επιτάχυνση τους, βασισμένοι σε *a priori* γνώση της θέσης τους στα προηγούμενα frames. Η χρησιμότητα του object tracking είναι σημαντική σε πλήθος εφαρμογών, όπως

- Video Animation : Σε ταινίες ή άλλης μορφής video είναι χρήσιμη ή αντικατάσταση ενός αντικειμένου με κάποιο άλλο (πχ ο κασκαντέρ σε μια επικίνδυνη σκηνή θα αντικατασταθεί με το πρωταγωνιστή. Μέσω object tracking μπορεί να γίνει ομαλά, αφού εισάγουμε το νέο αντικείμενο χωροχρονικά εκεί ακριβώς που ταιριάζει

- Surveillance Systems : Το «κλείδωμα» και η περαιτέρω παρακολούθηση ύποπτων αντικειμένων (πχ σύστημα σε μια τράπεζα) ή γενικά αντικειμένων με ενδιαφέρον (πχ μέτρηση ταχύτητας και θέσης αυτοκινήτων στο Εθνικό Δίκτυο Κυκλοφορίας) αποτελεί ακόμα μια σημαντική εφαρμογή.

- Μετεωρολογικά/ Περιβαλλοντικά συστήματα: Με την παρακολούθηση της μετακίνησης πχ ενός τυφώνα ή του μεγέθους μιας πετρελαιοκηλίδας

- Βαλλιστικά συστήματα/ Παρακολούθηση Στόχων: Αποτελεί ίσως το πεδίο με τις περισσότερες πρακτικές εφαρμογές. Θέλουμε να γνωρίζουμε την όσο το δυνατόν ακριβή θέση και ταχύτητα του στόχου ώστε πχ στην επόμενη χρονική slot να κατευθύνουμε την αεράμυνα στη θέση που θα πάει ή να εντοπίσουμε τη θέση που θα πέσει ένας πύραυλος ( forward tracking ) ή τη θέση που εκτοξεύτηκε (backward tracking).

Αλλά και πλήθος άλλων ανάλογων εφαρμογών.

Όπως και με το change detection, υπάρχουν μια σειρά προβλημάτων που πρέπει να ληφθούν υπόψη, και πάντα με βάση τις απαιτήσεις, ώστε να έχουμε το επιθυμητό αποτέλεσμα. Προβλήματα θορύβου (λόγω φωτισμού, κάμερας και κίνησης αυτής κτλ), μη κινούμενων αντικειμένων που συγχέονται με το στόχο, γενικά τα προβλήματα που είδαμε στο change detection, εμφανίζονται και εδώ καθιστώντας λογικό το change detection ως πρώτο στάδιο της διαδικασίας. Ακόμα, προβλήματα που έχουν να κάνουν με το «κλείδωμα» του στόχου δηλ. απότομες μεταβολές στη κίνηση, απουσία επαρκών δεδομένων κτλ, αυτά αποτελούν πρόβλημα του κυρίως αλγορίθμου.

Γενικά, το tracking είναι καθαρά ένα πρόβλημα προσεγγίσεων. Η μέχρι τώρα μεθοδολογία στηρίζεται κατά βάση στη θεώρηση κάποιων μοντέλων κίνησης για το αντικείμενο (πχ κίνηση με σταθερή ταχύτητα ή σταθερή επιτάχυνση). Προσπαθεί έπειτα με βάση προηγούμενες μετρήσεις να καταλήξει σε μια *a priori* πρόβλεψη της κίνησης στο frame  $n$ . Όταν η πραγματική θέση στο  $n$  του γίνεται γνωστή περνά έπειτα σε μια διαδικασία διόρθωσης της πρόβλεψης ώστε να προσεγγίζει τη πραγματική θέση. Αυτό το κάνουμε, και δεν πχ θέτουμε την πραγματική θέση αυτή ως διόρθωση, γιατί ακριβώς θέλουμε να προσεγγίσουμε ένα συνολικό μοντέλο κίνησης και όχι απλώς για κάθε λάθος πρόβλεψη να ακολουθούμε το αντικείμενο.

Έπειτα, από τη διόρθωση αυτή, λαμβάνει χώρα μια πρόβλεψη για το  $n+1$  κτλ.



Παρακάτω, παρουσιάζουμε μια σειρά τέτοιων αλγορίθμων, την οικογένεια των ονομαζόμενων α-b (ή α-b-k ) φίλτρων – τις πλέον χρησιμοποιούμενες μεθόδους tracking – καταλήγοντας στον πλέον χαρακτηριστικό και αξιόπιστο εκπρόσωπο τους, τα φίλτρα kalman και τις μορφές τους. Στο τμήμα της δική μας εργασίας (κεφ. 5) αναλύουμε τις απαιτήσεις μιας προσέγγισης tracking στα video που έχουμε και υλοποιούμε ένα φίλτρο kalman ελέγχοντας το αποτέλεσμα.

Εδώ, αν και βρήκαμε διάφορες πηγές ουσιαστικά στηριχθήκαμε στο [36], αφού εδώ βρίσκεται συγκεντρωμένη **όλη αυτή η πληροφορία για α-b filters και kalman**, που συναντήσαμε στις διάσπαρτες άλλες πηγές.

## 3.2 ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΛΟΓΙΚΗ

Έστω ένα αεροσκάφος που τη χρονική στιγμή n-1 το στίγμα του στο Radar (δεν έχω αυτή καθαυτή εικόνα αλλά το παράδειγμα είναι αρκετά κατατοπιστικό) είναι σε απόσταση 1200 μέτρα και οι ταχύτητα του 1000 m/sec. Θέλω από αυτά τα στοιχεία να κάνω μια πρόβλεψη της θέσης του τη χρονική στιγμή n. Σε αυτές τις εφαρμογές είναι λογικό να υποθέσω ότι η κίνηση του αεροπλάνου ακολουθεί το παρακάτω μοντέλο,

$$\begin{aligned}x_{n+1} &= x_n + T\dot{x}_n \\ \dot{x}_{n+1} &= \dot{x}_n\end{aligned}\quad (1)$$

τα αεροπλάνα σχεδόν αποκλειστικά κινούνται σε σταθερές τροχιές και μεταβάλλουν ελάχιστα την ταχύτητα τους, άρα υποθέτω αρχικά σταθερή ταχύτητα (ίδια με τη στιγμή n-1 και η νέα θέση η παλιά συν αυτή που θα πήγαινε με τη σταθερή αυτή ταχύτητα).



### ΕΙΚΟΝΑ 3.2.1 TRACKING

Συνεπώς, η νέα θέση του αεροπλάνου προβλέπουμε να είναι απόσταση  $1200 + 1 \cdot 1000 \text{ m} = 1200 \text{ m}$   
ταχύτητα 1000 m/sec

όπου T ο χρόνος που μεσολαβεί μεταξύ των δύο μετρήσεων (έστω εδώ 1 sec).

Έστω τώρα παίρνουμε την νέα μέτρηση του radar για τη n χρονική στιγμή και βλέπουμε ότι αυτή δίνει απόσταση 1290 m και ταχύτητα 1060m. Τι έγινε λοιπόν; Κάναμε λάθος υπόθεση ή όχι; Τελικά επιτάχυνε/ επιβράδυνε/ παρέμεινε σταθερή η ταχύτητα του αεροπλάνου; Ποια είναι τελικά η θέση του αεροπλάνου; Οι απαντήσεις των παραπάνω πέφτουν κάπου στη μέση. Είναι λογικό το radar να δίνει μετρήσεις με κάποιο σφάλμα, ανάλογα του τύπου και της ποιότητας κατασκευής του αυτό μπορεί να είναι αρκετά σημαντικό ή όχι. Καταλήγουμε σε κάποια μέση λύση για τη τελική μας πρόβλεψη

$$\dot{x}_{n,n}^* = \dot{x}_{n,n-1}^* + h_n \left( \frac{y_n - x_{n,n-1}^*}{T} \right)$$

$$x_{n,n}^* = x_{n,n-1}^* + g_n (y_n - x_{n,n-1}^*) \frac{1}{T} \quad (2)$$

Δηλ. η θέση τελικά που αποφασίζουμε ότι βρίσκεται πραγματικά

ΤΕΛΙΚΗ ΑΠΟΣΤΑΣΗ N= ΠΡΟΒΛΕΨΗ ΑΠΟΣΤΑΣΗΣ N ΣΤΟ N-1 + b \*(ΣΤΙΓΜΑ N – ΠΡΟΒΛΕΨΗ ΑΠΟΣΤΑΣΗΣ N ΣΤΟ N-1

ΤΕΛΙΚΗ ΤΑΧΥΤΗΤΑ N= ΠΡΟΒΛΕΨΗ ΤΑΧΥΤΗΤΑΣ N ΣΤΟ N-1 +a\*(Y- C)

ΜΕ Y= ΤΑΧΥΤΗΤΑ ΠΟΥ ΠΡΟΚΥΠΤΕΙ ΑΠΟ ΤΗΝ ΤΕΛΕΙΩΣ ΕΥΘΕΙΑ ΚΙΝΗΣΗ ΓΙΑ Τ ΣΤΗ ΝΕΑ ΘΕΣΗ ΠΟΥ ΔΙΝΕΙ ΤΟ RADAR

ΚΑΙ C=ΤΑΧΥΤΗΤΑ ΣΕ ΕΥΘΕΙΑ ΓΙΑ Τ ΧΡΟΝΟ ΣΕ ΑΠΟΣΤΑΣΗ ΙΣΗ ΜΕ ΤΗΝ ΠΡΟΒΛΕΨΗ ΑΠΟΣΤΑΣΗΣ N ΣΤΟ N-1

Δηλ. με απλά λόγια αποφασίζουμε ότι η αλήθεια είναι κατά ένα ποσοστό μεταξύ της πρόβλεψης και της μέτρησης του Radar.

Αν τώρα, ενδιαφερόμαστε για την πρόβλεψη στην επόμενη θέση n+1

Ο συνδυασμός της (1) με την (2) μας δίνει

$$\dot{x}_{n+1,n}^* = \dot{x}_{n,n-1}^* + \frac{h_n}{T} (y_n - x_{n,n-1}^*)$$

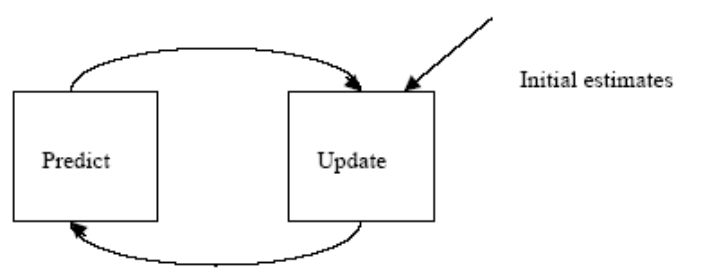
$$x_{n+1,n}^* = x_{n,n-1}^* + T \dot{x}_{n+1,n}^* + g_n (y_n - x_{n,n-1}^*) \quad (3)$$

Αυτό είναι ένα ενδιάμεσο στάδιο μιας διαδικασίας tracking, διόρθωση της πρόβλεψης για n, πρόβλεψη στο n+1. Τα α,b (h-g) δεν είναι απαραίτητα σταθερά, στα φίλτρα kalman μεταβάλλονται σε κάθε πρόβλεψη. Η διαδικασία ξεκινά από κάποια αρχική κατάσταση (αρχικοποίηση φίλτρου) για την οποία θα πούμε παρακάτω. Όμως αυτά είναι αρκετά για να χαρακτηρίσουν μια ολόκληρη οικογένεια αλγορίθμων πρόβλεψης. Έχουμε λοιπόν εδώ τα α,b φίλτρα, διαδικασίες πρόβλεψης γραμμικών συστημάτων, συγκεκριμένα εδώ του object tracking σε γραμμική κίνηση

- 
1. Wiener filter
  2. Fading-memory polynomial filter
  3. Expanding-memory (or growing-memory) polynomial filter
  4. Kalman filter
  5. Bayes filter
  6. Least-squares filter
  7. Benedict-Bordner filter
  8. Lumped filter
  9. Discounted least-squares g-h filter
  10. Critically damped g-h filter
  11. Growing-memory filter
- 

### EΙΚΟΝΑ 3.2.2 α-b filters

Αυτό που τα διαφοροποιεί είναι ο τρόπος εύρεσης των α-b και το initialization ενώ κάποια στη πράξη είναι ταυτόσημα ή ελαφρώς διαφοροποιούμενη μορφή το ένα του άλλου. Παρατηρούμε λοιπόν πως η βασική τους λογική βασίζεται στην εξής ροή



**ΕΙΚΟΝΑ 3.2.3 PREDICTION-UPDATE MODEL**

Σε κάθε βήμα, κάνουμε μια πρόβλεψη με βάση κάποιο μοντέλο (εδώ κίνησης ) και διορθώνουμε τη πρόβλεψη με βάση κάποια διαδικασία. Μετά προβλέπουμε και διορθώνουμε για την επόμενη χρονική στιγμή – πάντα με βάση τα τελικά αποτελέσματα της προηγούμενης κτλ. Το συγκεκριμένο μοντέλο ονομάζεται δίκτυο bayes. Χαρακτηρίζουμε τις μεθόδους αυτές «Φίλτρα» γιατί «φιλτράρουμε»/διορθώνουμε την αρχική πρόβλεψη. Η χρήση τους είναι ευρεία όχι μόνο για tracking αλλά και αλλού (πχ το φίλτρο wiener είναι κλασσική μέθοδο αποθορυβοποίησης εικόνας με salt n’ pepper noise). Παρακάτω, αναλύουμε το φίλτρο kalman, αρκούμαστε σ’ αυτό γιατί όπως θα δείξουμε είναι πληρέστατο και υπερέχει όλων των άλλων όμορων φίλτρων α-b.

### 3.3 ΦΙΛΤΡΑ KALMAN

#### 3.3.1 ΟΡΙΣΜΟΣ

Τα φίλτρα kalman γενικά πάνε ένα μικρό βήμα παραπάνω το tracking από αυτά που είπαμε ως τώρα. Με το μοντέλο της σταθερής ταχύτητας που ορίσαμε πριν, σε περιπτώσεις μεταβολής της ταχύτητας το φίλτρο αργεί να ανταποκριθεί και να προσαρμόσει την ταχύτητα που βρίσκει στην πραγματική. Ο kalman για να αυξήσει το response time ανακαθόρισε τις εξισώσεις του predict βήματος ως εξής

$$\begin{aligned} x_{n+1} &= x_n + T\dot{x}_n \\ \dot{x}_{n+1} &= \dot{x}_n + u_n \end{aligned} \quad (4)$$

δηλ. μια μεταβλητή  $u$  ορίζει αυτήν τη τυχαία μεταβολή στη ταχύτητα, το τυχαίο jump σε μια νέα τιμή, όταν συμβαίνει αυτή η μεταβολή.

Ας θεωρήσουμε τώρα τη πραγματική θέση του αντικειμένου  $x_{n+1}$  και η πρόβλεψη  $x_{n+1,n}^*$ . Ο kalman εξέτασε τη δυνατότητα ελαχιστοποίησης του σφάλματος

$$(x_{n+1,n}^* - x_{n+1})^2 \quad (5) \text{ (mean square error)}$$

μετά από μελέτες κατέληξε στην παρακάτω βέλτιστη λύση

$$\begin{aligned}\dot{x}_{n+1,n}^* &= \dot{x}_{n,n-1}^* + \frac{h_n}{T}(y_n - x_{n,n-1}^*) \\ x_{n+1,n}^* &= x_{n,n-1}^* + T\dot{x}_{n+1,n}^* + g_n(y_n - x_{n,n-1}^*)\end{aligned}\quad (6)$$

που δεν είναι άλλος από το γενικό τύπο (3) που διαισθητικά παρουσιάσαμε παραπάνω. Για το φίλτρο kalman τα g,h αλλάζουν σε κάθε βήμα. Όπως θα δείξουμε παρακάτω, τα g,h είναι συναρτήσεις της διασποράς της διαφοράς μεταξύ της μέτρησης που δίνει το radar/στοιχείο αρχικής μέτρησης και της πραγματικής θέσης του αντικειμένου. Στην κατάσταση ηρεμίας, όπου το φίλτρο έχει σταθερό σφάλμα – απόκλιση μεταξύ των δύο αυτών μετρήσεων τότε ισχύει

$$h = \frac{g^2}{2 - g} \quad (7)$$

δηλ. το h τότε σχετίζεται το g. [36.iii]

### 3.3.2 ΣΤΟΙΧΕΙΑ ΥΠΟΛΟΓΙΣΜΟΥ ΦΙΛΤΡΩΝ KALMAN

Εδώ, υπολογίζουμε τις παραμέτρους g-h και κάνουμε μια πρώτη παρουσίαση του γιατί τα kalman είναι η βέλτιστη tracking λύση για το μοντέλο που παρουσιάσαμε. Όπως είπαμε πριν, τα g-h φίλτρα επιχειρούν να συνδυάσουν δύο μετρήσεις-estimations. Την  $y_n$  και την πρόβλεψη  $x_{n,n-1}^*$ . Το kalman συνδυάζει τις δυο μετρήσεις αυτές ώστε να υπολογίσει μια filtered  $x_{n,n}^*$ , δηλ. μια διόρθωση της πρόβλεψης όταν πλέον είναι γνωστή η πραγματική θέση για τη στιγμή n, βασισμένη και στις δυο. Έχει υπολογιστεί [36.i] ότι η προσέγγιση με τη μεγαλύτερη ακρίβεια είναι αυτή με

$$x_{n,n}^* = \left[ \frac{x_{n,n-1}^*}{VAR(x_{n,n-1}^*)} + \frac{y_n}{VAR(y_n)} \right] \frac{1}{\left( \frac{1}{VAR(x_{n,n-1}^*)} + \frac{1}{VAR(y_n)} \right)} \quad (8)$$

όπου VAR η διασπορά κάθε μεταβλητής.

Αρα το (8) παρέχει μια καλή προσέγγιση που συνδυάζει τις δυο μετρήσεις. Από την (8) τώρα, εύκολα μπορούμε να εξάγουμε το

$$x_{n,n}^* = x_{n,n-1}^* + \frac{VAR(x_{n,n-1}^*)}{VAR(y_n)}(y_n - x_{n,n-1}^*) \quad (9) \text{ δηλ.}$$

$$g_n = \frac{VAR(x_{n,n-1}^*)}{VAR(y_n)} \quad (10)$$

και ανάλογα

$$h_n = \frac{COV(x_{n,n}^* \dot{x}_{n,n}^*)}{VAR(y_n)} \quad (11)$$

έχουμε λοιπόν, όλες τις εξισώσεις που χρειαζόμαστε για να υλοποιήσουμε ένα φίλτρο kalman. Μας απομένει μόνο το initialisation και πως περνάμε σε 2-D κίνηση.

Για το initialization συνήθως εμπιστευόμαστε τις 2 πρώτες μετρήσεις (θα εξηγήσουμε περαιτέρω στην υλοποίηση) για το 2<sup>ο</sup> απλώς εφαρμόζω δύο kalman filters, ένα για χ-άξονα και ένα για y-άξονα, θεωρώντας τις δύο μετακινήσεις x,y ασυσχέτιστες (στην πράξη αυτό δεν ισχύει αλλά συνήθως το χρησιμοποιούμε διότι δεν δίνει μεγάλη απόκλιση από την πραγματικότητα και απλοποιεί τις πράξεις). Οι υπόλοιπες λεπτομέρειες αναλύονται στο κεφ.5 της υλοποίησης.

Συνεπώς, αποκρυσταλλώνοντας τον αλγόριθμο έχουμε

- Για κάθε N+1 frame το prediction δίνει το  $x_{n+1,n}^*$

$$\dot{x}_{n+1,n}^* = \dot{x}_{n,n-1}^* + \frac{h_n}{T} (y_n - x_{n,n-1}^*)$$

$$x_{n+1,n}^* = x_{n,n-1}^* + T \dot{x}_{n+1,n}^* + g_n (y_n - x_{n,n-1}^*)$$

$$\text{Με } g_n = \frac{VAR(x_{n,n}^*)}{VAR(y_n)}, \text{ όπου}$$

$$VAR(x_{n,n}^*) = E(\{x_{n,n}^* - E(x_{n,n}^*)\}^2) = \sigma_{\chi_{n,n}^*}^2$$

$$VAR(y_n) = \sigma_{y_n}^2$$

$$\text{Όπου } h_n = \frac{COV(x_{n,n}^* \dot{x}_{n,n}^*)}{VAR(y_n)}, \text{ με γενικά } COV(AB) = A^T B$$

yh= Η μέτρηση-στίγμα του χ (θα δούμε πως μπορούμε να έχουμε κάτι τέτοιο στο δικό μας σύστημα)

με γενικό δεδομένο ότι κάθε  $x_{n,n}^*$  προκύπτει από το

$$\dot{x}_{n,n}^* = \dot{x}_{n,n-1}^* + h_n \left( \frac{y_n - x_{n,n-1}^*}{T} \right)$$

$$x_{n,n}^* = x_{n,n-1}^* + g_n (y_n - x_{n,n-1}^*) \frac{1}{T}$$

Για να περάσω σε 2-D τώρα, θέτω την κίνηση στους 2 άξονες όπως είπαμε ασυσχέτιστες (δεν ισχύει στη πράξη, αλλά η προσέγγιση αυτή είναι διαδεδομένη, καθώς απλοποιεί τη διαδικασία ενώ δεν αποκλίνει πολύ συνήθως από την πραγματική κίνηση) και άρα κάνω χρήση 2 διαφορετικών kalman filter, ένα για κάθε άξονα. Όσο για τον παράγοντα  $u_n$ , μια καλή προσέγγιση [36.ii] είναι να θέσω  $u_n = 3\text{VAR}(y_n)$ .

### 3.3.3 ΓΙΑΤΙ KALMAN FILTER

Γενικά, η οικογένεια αυτή των α-b filters είναι ένα πολύ ισχυρό εργαλείο πρόβλεψης. Ιδιαίτερα καλά αποτελέσματα τα φίλτρα αυτά επιτυγχάνουν σε γραμμικά προβλήματα, όπως πχ εδώ που υποθέτω η μετακίνηση του αντικείμενου είναι γραμμική συνάρτηση της μετακίνησης στο προηγούμενο frame (1) (γι' αυτό και οι εφαρμογές σε airplane surveillance κτλ είναι διαδεδομένες). Το kalman filter υπερτερεί των υπολοίπων α-b filters στο ότι ακριβώς είναι πιο γενικό, μας παρέχει μια πλήρη διαδικασία εύρεσης g-h, δεν τα θεωρεί πχ σταθερά, και γενικά αποτελεί τη γενικότερη μορφή α-b filter, σχεδόν όλα τα άλλα α-b είναι εφαρμογή kalman με μια υπόθεση που απλοποιεί κάποια δεδομένα κτλ

Ακόμα, ως ξαναδούμε το αρχικό μοντέλο κίνησης που θεωρεί το kalman

$$x_{n+1} = x_n + T\dot{x}_n$$

$$\dot{x}_{n+1} = \dot{x}_n + u_n$$

Στη βάση της, η θεωρία δεν μας λέει πώς να βρούμε επακριβώς το  $u_n$ , που δεν είναι τίποτα άλλο από μια τυχαία μεταβολή της ταχύτητας, σε περίπτωση που έχω μεταβολή. Η λύση που επιλέγεται γενικά είναι η χρήση μιας απλής σταθεράς (πχ 5) ως διασπορά της  $u$ . Στο [36.ii], που αποτέλεσε και την κυρίως βάση ανάγνωσης για το tracking, η λύση που επιλέγεται, και που επιλέγουμε και εμείς είναι  $u_n = \sigma \text{VAR}(X_{n,n}) y_n$  όπου  $\sigma = 3(\text{Var}(X_{n,n}))$ , που είναι μια πρακτική προσέγγιση, γενικά με καλά αποτελέσματα, ιδιαίτερα σε radar που είναι και το βασικό αντικείμενο του βιβλίου.

Το 2<sup>ο</sup> σκέλος στη πράξη μας λέει βασικά ότι η πρόβλεψη ταχύτητας δεν είναι τίποτα άλλο παρά μια τυχαία μεταβλητή + την προηγούμενη ταχύτητα = **μια τυχαία μεταβλητή**. Δηλ. αυτή ακριβώς είναι η μεγαλύτερη αξία του kalman filter, **μπορούμε αργότερα με κάποιες μορφές μοντελοποίησης να βελτιώσουμε ίσως τα αποτελέσματα (πχ με υπόθεση γκαουσιανής στο  $u$  όπως στο change detection)**. Δηλ. θεωρώντας διαφορετικό μοντέλο πχ για την διόρθωση της Ταχύτητας  $X_{n,n}$  να καταλήξουμε σε καλύτερη απόδοση και... **σε μια νέα μορφή α-b kalman filter**. Ενώ στα άλλα α-b, τα αποτελέσματα είναι συγκεκριμένα, το πολύ να μπορείς να συνδυάσεις δύο τύπους α-b filters. Με αυτή τη λογική έχουν αναπτυχθεί μια σειρά βελτιώσεων με χρήση πχ EM (estimation minimization) κτλ [37].

### 3.4 EKF – EXTENDED KALMAN FILTER

Αν και ο παραπάνω ορισμός του kalman filter δίνει καλά αποτελέσματα και ειδικά σε radar/ airplane / missile κτλ surveillance είναι μια ευρύτατα χρησιμοποιούμενη μέθοδος, ωστόσο υπάρχουν κάποια προβλήματα που η παραπάνω μεθοδολογία δεν δύναται να λύσει, τουλάχιστον με υψηλό ποσοστό επιτυχίας.

Στο προηγούμενο, γραμμικό μοντέλο κίνησης, θεωρούμε πως οι παρατηρήσεις  $y_n$  και οι προβλέψεις  $x_n$ , συνδέονται ως εξής

$$y_n = x_n + v_n \quad (3.4.1)$$

όπου  $v_n$  η απόκλιση μεταξύ αυτών. Δηλ. θεωρούμε πως παρατήρηση και πρόβλεψη έχουν γραμμική σχέση. Αυτό, αν και είναι μια καλή αρχική προσέγγιση σε περιπτώσεις όπως πχ ακανόνιστη κίνηση ανθρώπων δεν ισχύει.

$$\text{Άρα τότε, } y_n = f(x_n) + v_n \quad (3.4.2)$$

Όπου  $f$ , μια μη γραμμική συνάρτηση.

Σε αυτή λοιπόν την περίπτωση όπου έχω μη γραμμική κίνηση (απότομες εναλλαγές στην ταχύτητα και στροφές στην κίνηση), απαιτείται κάποια αλλαγή στη θεώρηση και στους υπολογισμούς του kalman filter. Αυτό βασικά που γίνεται είναι, σε γενικές γραμμές, η μετατροπή της  $f$  σε γραμμική συνάρτηση (μέσω ανάπτυξη σε σειρά Taylor, χρήση Least Square Approximations κτλ).

Εμείς δεν ασχοληθήκαμε με το κομμάτι αυτό, ούτως ή άλλως το κύριο βάρος στη διπλωματική δόθηκε στο change detection, για περισσότερα μπορεί κανείς να δει στο [36.iv]. Το μόνο που έχουμε κάνει σε αυτό το πεδίο είναι να ελέγξουμε ως ένα βαθμό κατά πόσο μπορεί το γραμμικό kalman να αποδώσει σε φαινόμενα μη γραμμικής κίνησης, όπως θα δούμε παρακάτω στα tests.

## **ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ CHANGE DETECTION**

- Παρουσίαση Των Videos, Προβλήματα Που Προκύπτουν
- Επιλογή Των Αλγορίθμων Change Detection Και Συνολικός Καθορισμός Των Evaluation Διαδικασιών
- Παρουσίαση Αποτελεσμάτων και Συμπερασμάτων



## 4.ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ CHANGE DETECTION

Με βάση τα παραπάνω, σε πρώτο στάδιο θελήσαμε να κάνουμε κάποια αρκετά διεξοδικά test σε μια σειρά από τους αλγόριθμους που συναντήσαμε. Αυτό όμως που κρατήσαμε περισσότερο είναι ότι το preprocessing πρέπει να προσεχθεί ιδιαίτερα και ότι πιθανών ( πράγμα που επιβεβαιώθηκε στη πράξη) να είναι αυτό που μπορεί να καθορίσει καίρια την απόδοση των αλγορίθμων.

Ξεκινάμε λοιπόν ορίζοντας το τι θέλουμε να κάνουμε. Θέλουμε λοιπόν με βάση δύο videos που έχουμε να αναπτύξουμε ένα σύστημα change detection και tracking το οποίο να αποδίδει καλά και στα δύο videos, τα οποία είναι εντελώς διαφορετικά ως προς το περιβάλλον τους. Απομένει λοιπόν σε μας να επιλέξουμε το preprocessing και τους αλγόριθμους που θα δώσουν τέτοιο αποτέλεσμα. Οι επιλογές που μπορούμε να κάνουμε είναι πάρα πολλές. Έτσι, πρέπει αφενός μεν μέσω της γνώσης που αποκομίσαμε με την θεωρητική έρευνα μας και την εμπειρία να απορρίψουμε κάποιες και αφετέρου, μέσω ελέγχου να επιλέξουμε από τις υπόλοιπες το κατά το δυνατόν καλύτερο.

### 4.1 ΚΡΙΤΙΚΗ ΤΩΝ VIDEOS

Έχουμε λοιπόν δυο videos

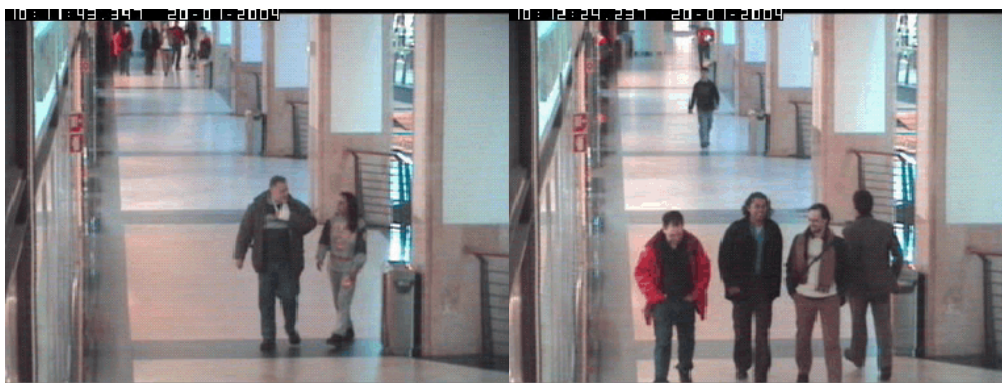
1)



Το πρώτο, 4:50 λεπτά διάρκεια (κίνηση ανθρώπων σε ημυπαίθριο διάδρομο, 624 frames σε DIVX 4.12 δηλ. σχετικά χαμηλή κίνηση, 384X288, 24-bit RGB) στο οποίο έχω αρκετό θόρυβο λόγω κάμερας («χιόνια» εμφανίζονται, κυρίως γύρω από κινούμενα αντικείμενα), με σχετικά «χαμηλή» κίνηση και μικρά κινούμενα αντικείμενα, στο οποίο έχω ισχυρά επίδραση εξωτερικού ακανόνιστου φωτός ( το φως από τη τζαμαρία και η αντανάκλαση του). Επιπλέον, ανά 3-5 frames, έχω flickering στην εικόνα.

Συνεπώς στο video αυτό πρέπει οπωσδήποτε να κάνω προεπεξεργασία για αφαίρεση του εξωτερικού φωτός και δημιουργία illumination invariant περιβάλλοντος, να λάβω υπόψη το flickering, που αυτό θεωρητικά σημαίνει πως οι αλγόριθμοι μας δεν πρέπει να αποδίδουν καλά ούτως ή άλλως (κινούμενη κάμερα) και παράλληλα να λάβω υπόψη ότι ο θόρυβος είναι μεγάλος αλλά και τα αντικείμενα μικρά, συνεπώς θα επηρεάζει αρκετά και θα δίνει false changes στα difference images.

2)



1:05 λεπτά διάρκεια, (κίνηση ανθρώπων σε εσωτερικό διάδρομο, 1650 frames σε DIVX 4.12 δηλ. εντονότερη από το προηγούμενο, 384X288, 24-bit RGB). Τα αντικείμενα εδώ είναι περισσότερα, κινούνται πιο πολύ και περιπλέκονται, αυξομειώνεται το μέγεθος τους (αφού κινούνται από/ προς τη θέση της κάμερας). Παράλληλα και εδώ έχω θόρυβο (χιόνι αλλά και «σπασίματα» των regions) λόγω κάμερας ενώ παρατηρείτε ότι ανά 5-8 frames η εικόνα «παγώνει». Αυτό δεν επηρεάζει τόσο το change detection. Βέβαια, ενδιαφέρον παρουσιάζει ο έλεγχος του αποτελέσματος του change detection μεταξύ των «παγωμάτων» αυτών, αφού τότε το αποτέλεσμα πρέπει (αν κάνω σύγκριση frame με το προηγούμενο του) να δίνει κενό, εξαιρώντας την επίδραση θορύβου κτλ). Επηρεάζει όμως αρκετά το tracking, αφού θα δημιουργεί ασυνέχειες στη συνάρτηση κίνησης των αντικειμένων.

## 4.2 ΑΛΓΟΡΙΘΜΟΙ CHANGE DETECTION/ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΠΟΥ ΕΠΙΛΕΧΘΗΚΕ

Ξεκινώντας λοιπόν, απομένει να ορίσουμε τη στρατηγική μας, στο πρώτο στάδιο, στο οποίο τεστάρουμε change detection αλγόριθμους και διαδικασίες preprocessing, για να επιλέξουμε τη βέλτιστη για εμάς.

### 4.2.1 ΕΠΙΛΟΓΗ ΠΡΟΕΠΕΞΕΡΓΑΣΙΑΣ

Από τα videos βλέπω ότι και στα δύο θέλω να αφαιρέσω τον πρόσθετο εξωτερικό φωτισμό, ειδικά στο 1<sup>ο</sup> η επίδραση του είναι ισχυρή. Επίσης, θέλω να ελέγξω κατά πόσο μπορώ να αφαιρέσω θόρυβο κάμερας με το φίλτρο για τον εξωτερικό φωτισμό. Θέλω βασικά να δω πόσο «ατονούν» οι διαφορές λόγω θορύβου της κάμερας με την αφαίρεση του φωτισμού από αυτές. Τέλος, εν γένει θέλω να δω, τη διαφορά επίδοσης των αλγορίθμων με και χωρίς τέτοια επεξεργασία. Για όλα αυτά, επιλέγω και υλοποιώ homomorphic filtering, με την διαδικασία που αναλύεται στο παράρτημα C.

Για τη μορφή των εικόνων, ερευνώ την επίδραση της ύπαρξης χρώματος ή όχι. Άρα, θα κάνω change detection τόσο σε επίπεδο grayscale, σε έγχρωμο 24-bit RGB, και φυσικά σε έγχρωμο 24-bit RGB με homomorphic preprocessing.

Για τη δημιουργία των difference images, επιλέγω στο στάδιο των tests απλή απόλυτη τιμή διαφοράς μεταξύ των  $I_v$  και  $I_{v-1}$  frames (δηλ. το frame μείον το προηγούμενο του). Με αυτό τον τρόπο, θα παρατηρήσω ποιοι αλγόριθμοι εξαλείφουν καλύτερα το θόρυβο της κάμερας (αφού τώρα ο θόρυβος συγκρίσιμος με αυτό που

ψάχνω σαν αλλαγή, εδώ την κίνηση), καθώς επίσης θα παρατηρήσω καλύτερα την «αυστηρότητα» των αλγορίθμων, δηλ. πόσο υπερεκτιμούν ή υποτιμούν τη τιμή του threshold.

#### 4.2.2 ΕΠΙΛΟΓΗ ΑΛΓΟΡΙΘΜΩΝ

Οι επιλογές μου είναι πάρα πολλές αλλά σαν κυρίως change detection αλγορίθμους θέλω κάτι σχετικά απλό και γρήγορο (ώστε πιθανών να μπορώ να μετατρέψω το σύστημα σε real-time αργότερα), ταυτόχρονα όμως με υψηλή επίδοση, γενικευμένο δηλ. να χρησιμοποιείτε αρκετά ή να είναι μέρος άλλων μεθόδων στη βιβλιογραφία (ώστε να μπορώ να ελέγξω τα αποτελέσματα μου με αυτά άλλων εργασιών). Με αυτό το σκεπτικό αφήνω σε αυτή την εργασία το adaptive μέρος στην άκρη ή τη μέθοδο Aach et al (αποτελεί συνδυασμό γενικών στοιχείων όπως homomorphic filtering, Gaussian change detection που υλοποιώ ούτως ή άλλως για όλη τη μελέτη) και επικεντρώνομαι σε

1) **Gaussian Noise Modeling**: Το core αυτού του αλγόριθμου ή η ίδια η μέθοδος αποτελεί το δημοφιλέστερο ίσως τρόπο επίλυσης ενός change detection προβλήματος. Ταυτόχρονα, είναι σχετικά εύκολος στην υλοποίηση του ( οι τρόποι εύρεσης της error complementary function erfc και της προσέγγισης της διασποράς του θορύβου εξηγούνται στα ανάλογα παραρτήματα ενώ ο υπόλοιπος αλγόριθμος είναι μια σειρά τετριμμένων πράξεων πάνω σε αυτά). Βέβαια, η απόδοση του επηρεάζεται και από «εμπειρικές» παραμέτρους όπως την ακρίβεια της τιμής που έχω ως διασπορά θορύβου, το σκαλοπάτι  $P_f$  κτλ. Γι' αυτά, θα εμπιστευτούμε τη γενική τάση στη βιβλιογραφία που βρήκαμε, δηλ. χρήση LEAST MEDIAN OF SQUARES για προσέγγιση της διασποράς θορύβου [4 και αλλού] και  $P_f=5\%$  που είναι αρκετά μικρό ώστε να μην έχω μεγάλη απώλεια χρήσιμης πληροφορίας αλλά και αρκετά μεγάλο ώστε να μην έχω υψηλό interference του θορύβου.

2) **Poisson Noise Modeling**: Συνήθως τη συναντούμε σε αντιπαραβολή με την Gaussian στη βιβλιογραφία και κρίνουμε και εμείς σκόπιμο να κάνουμε τον ίδιο έλεγχο. Σαν τη Gaussian, και αυτή έχει να κάνει με noise modeling, αλλά εδώ η μοντελοποίηση δε γίνεται σε αναφορά με το ίδιο το pixel αλλά σε σχέση με όλη τη γειτονιά του. Πρόβλημα αποτελεί το ότι εξαρτάται από το μέγεθος παραθύρου, δηλ. της γειτονιάς που θα ορίσω. Γι' αυτό, επιλέγω μέγεθος  $25 \times 25$ , από κάποιες πρώτες μικρές δοκιμές που έδειξαν ότι εκεί έχω καλύτερη απόδοση. Κατά τ' άλλα, αποτελεί μια σειρά στατιστικών μετρήσεων.

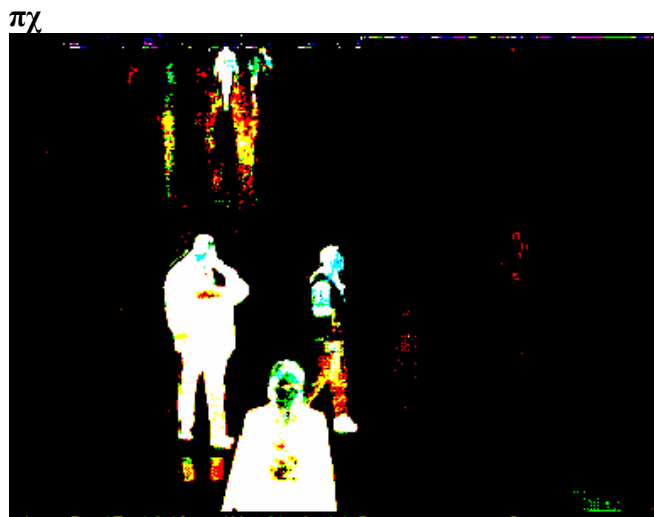
3) **Euler Number**: Εισάγει στην έρευνα το spatial modeling της ίδιας της πληροφορίας στην εικόνα και όχι του θορύβου, κάτι διαφορετικό με τα παραπάνω. Επίσης δεν εξαρτάται από καμιά άλλη παράμετρο ( δεν έχω παράθυρα κτλ εδώ)

#### 4.2.3 ΜΕΤΑΕΠΕΞΕΡΓΑΣΙΑ

Αν και οι επιλογές αυτού του σταδίου αφορούν το επόμενο κομμάτι του object tracking, αφού από τα αποτελέσματα που θα έχω εδώ θα επιλέξω μια μέθοδο και μετά η μεταεπεξεργασία θα γίνει ώστε να βελτιστοποιήσω τ' αποτελέσματα της, πρέπει ωστόσο να κάνω κάποια πράγματα ώστε το change detection αποτέλεσμα να είναι πιο εμφανές και συγκρίσιμο. Έτσι, επιλέγω να ελέγξω εδώ αν η χρήση median filtering στα image differences βελτιώνει το αποτέλεσμα [3].

Επιπλέον σε αυτά, πρέπει να επιλέξω πως θα συνδυάσω τ' αποτελέσματα των τριών καναλιών χρώματος R,G και B. Αν και υπάρχουν αρκετές επιλογές και εδώ (πχ μοντελοποίηση της συνεισφοράς κάθε καναλιού στην πραγματική πληροφορία με κάποια κατανομή, OR ή AND fusion κ.ά.) από πρώτες μικρές δοκιμές επιλέγω να κάνω μόνο AND fusion των καναλιών, δηλ.

**Αν στο pixel  $I(x)$  του image difference έχω και τα τρία κανάλια χρώματος 1 τότε και μόνο τότε θέτω το pixel σαν αλλαγή (1).**



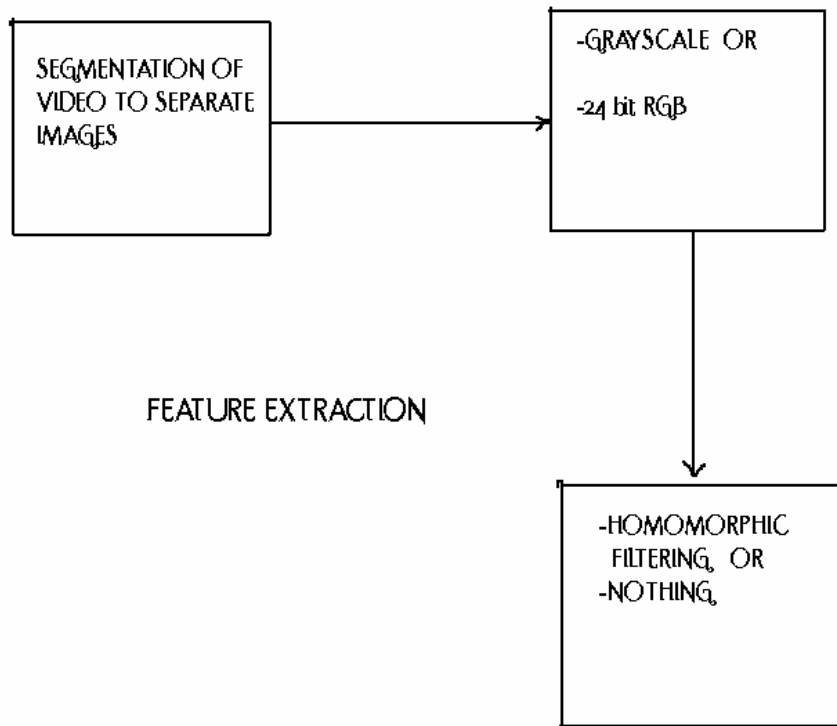
**ΕΙΚΟΝΑ 18 ΑΠΟΤΕΛΕΣΜΑ CHANGE DETECTION ΚΑΙ ΣΤΙΣ ΤΡΕΙΣ ΧΡΩΜΑΤΙΚΕΣ ΣΥΧΝΟΤΗΤΕΣ (ΕΔΩ EULER-COLOR + HOMOMORPHIC) (Η ΠΛΕΙΟΨΗΦΙΑ ΤΩΝ ΧΡΩΜΑΤΙΣΤΩΝ ΜΗ ΛΕΥΚΩΝ ΚΟΜΜΑΤΙΩΝ ΔΙΝΟΤΑΝ ΩΣ ΑΛΛΑΓΗ-ΛΕΥΚΟ- ΣΕ GRAYSCALE)**

Στην παραπάνω εικόνα βλέπω τους λόγους της επιλογής μας. Αλλαγές που γενικά θέλω να αποφύγω, όπως σκιές αντικειμένων και θόρυβος της κάμερας, έχουν τη τάση να εμφανίζονται περισσότερο σε συγκεκριμένες χρωματικές μπάντες και πολύ σπανιότερα και στις τρεις ενώ η μεγάλη πλειοψηφία των πραγματικών αλλαγών εμφανίζονται και στις τρεις χρωματικές μεταβλητές. Έτσι, επιλέγω αυτή τη προσέγγιση.

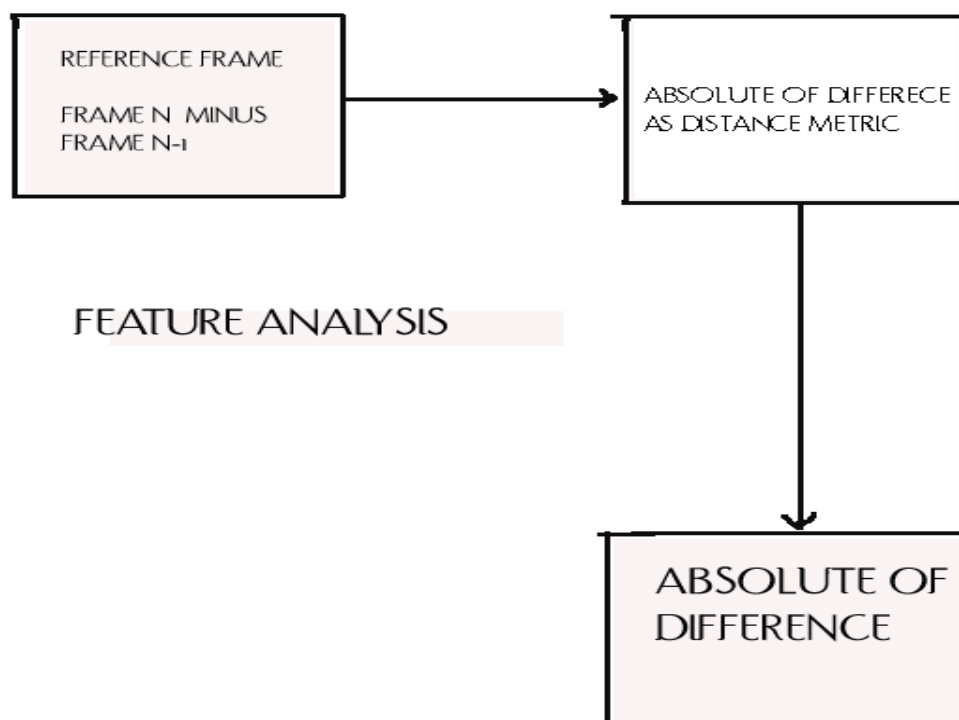
#### **4.2.4 ΕΠΙΛΟΓΟΣ**

Τελικά, έχω 18 συνολικά αρκετά χαρακτηριστικά συστήματα change detection τα οποία θα συγκρίνω και θα επιλέξω αυτό με το καλύτερο αποτέλεσμα. Σχηματικά με βάση τη 4-step μεθοδολογία που αναφέραμε στην αρχή

## 1) FEATURE EXTRACTION STEP



## 2) FEATURE ANALYSIS

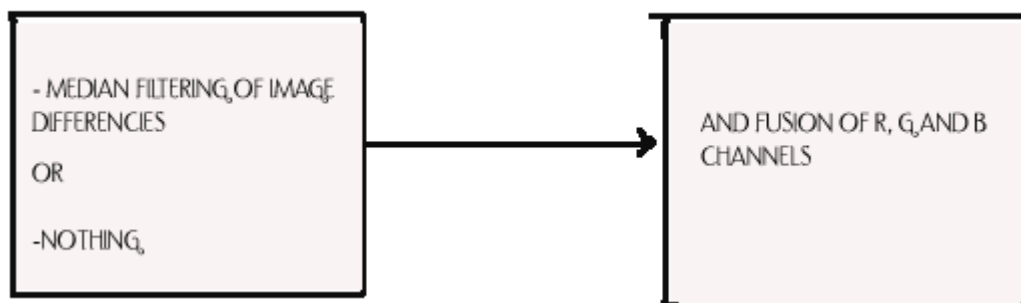


### 3) CLASSIFICATION

#### CLASSIFICATION

- GAUSSIAN MODELING, OR
- POISSON MODELING, OR
- EULER SHAPE DESCRIPTOR MODELING,

### 4) POST- PROCESSING



Όπου, διαδοχικά κάθε στάδιο τροφοδοτεί το αποτέλεσμα του ως είσοδο στο επόμενο

Τα παρακάτω ισχύουν και για το tracking μέρος της εργασίας

-Τα video (λόγω ασυμβατότητας της μορφής τους με τους video players, όπως μας παρουσιάστηκε) μετατράπηκαν σε **DIVX 4.12** μέσω του freeware converter **virtualdub** ([sourceforge.net](http://sourceforge.net))

-Η μετατροπή των videos σε frames (24-bit color bmp image segmentation) έγινε με το freeware converter **bink and smacker** ( [www.radgametools.com](http://www.radgametools.com))

-Για το άνοιγμα των αρχείων (προσπέλαση τιμής κάθε pixel κτλ) και το interaction γενικά χρησιμοποιήσαμε τη freeware C++ βιβλιοθήκη **coolimage** ([cimg.sourceforge.net](http://cimg.sourceforge.net))

**Όλα τα υπόλοιπα υλοποιήθηκαν από εμάς σε C++ , χρησιμοποιήσαμε VISUAL C++ 6.00, αλλά όλοι οι κώδικες είναι γενικοί και portable**

-Οι μετρήσεις έγιναν σε **P4 2.8 MHZ, 196 MB RAM**

## 4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

### 4.3.1 ΚΡΙΤΗΡΙΑ ΟΡΙΣΜΟΥ ΤΩΝ TEST

Όπως είδαμε στην εισαγωγή, ο ορισμός αξιόπιστων μετρικών σύγκρισης των μεθόδων change detection δεν είναι εύκολη υπόθεση. Χαρακτηριστικό είναι ότι στην ως τώρα βιβλιογραφία που συναντήσαμε (τόσο στις εργασίες του Rosin, του Cavallaro, του Aach και αλλού), η αποτίμηση των αποτελεσμάτων γίνεται περισσότερο με την οπτική επισκόπηση. Ωστόσο, υπάρχουν κάποιες μετρικές οι οποίες με τον υπολογισμό τους πάνω στις change masks δίνουν κάποια σημαντικά συγκριτικά στοιχεία

1) **Object metric X του Euler Number:** Προφανώς η μετρική αυτή δεν αποτελεί αξιόπιστη μέτρηση του αριθμού των Objects/Regions στην εικόνα ( το pattern που ορίζει ένα object εδώ μπορεί εύκολα να παρουσιάζεται πλέον της μιας φορές σε μια πραγματική region), αλλά είναι ένα αρκετά καλό σχετικό κριτήριο. Για την ίδια εικόνα, επεξεργασμένη με διαφορετικό Change Detection αλγόριθμο, το αποτέλεσμα που οπτικά δίνει τις περιοχές που θέλω και έχει λιγότερο X, είναι προφανώς καλύτερο γιατί πχ έχει λιγότερα noise regions (που συνεισφέρουν στο X), λιγότερες «κομμένες» regions, όχι **υποεκτίμηση του threshold** κτλ.

2) **Hole metric V του Euler Number:** Τα ίδια με το X ισχύουν εδώ περί αξιοπιστίας. Σε σχετικό βαθμό όμως, πάλι ένα καλό κριτήριο. Για την ίδια εικόνα, επεξεργασμένη με διαφορετικό Change Detection αλγόριθμο, το αποτέλεσμα που οπτικά δίνει τις περιοχές που θέλω και έχει λιγότερο V είναι καλύτερο διότι έχει σίγουρα λιγότερα holes και άρα η εσωτερική συνοχή της region είναι καλύτερη, δεν έχω χάσιμο πληροφορίας, όχι **υπερεκτίμηση του threshold** κτλ

3) **EULER NUMBER:** Από τα παραπάνω, και σε αναφορά με τα video που έχω, μια καλή change detection μέθοδο θα έχει σχετικά χαμηλό Euler number, που όμως δεν θα τείνει σε πάρα πολύ χαμηλές τιμές ( προς το 0 ή αρνητικό, δηλ. πολλές τρύπες στο αντικείμενο)

4) **Compactness:** Όσο πιο χαμηλό το κριτήριο αυτό, όπως το ορίσαμε στη θεωρία, τόσο πιο smoothed η region, τόσο πιο «καλοσχηματισμένη» είναι.

#### 5) ΑΡΙΘΜΟΣ ΤΩΝ REGIONS ΟΠΩΣ ΠΡΟΚΥΠΟΥΝ ΑΠΟ LABELING:

Θέλω να μην είναι μεγάλος ο αριθμός, άρα μεγαλύτερη συνοχή στα regions

Όλα αυτά, οι διαφορές στις μετρικές αυτές μεταξύ των change mask images, γίνονται ακόμα πιο έντονες με την επιλογή μας να μην κάνουμε το image difference ΕΙΚΟΝΑ minus Background αλλά Εικόνα minus Προηγούμενη της Στη διαδοχή του Video, τώρα τα regions είναι πιο μικρά ( κίνηση ), πιο μεταβαλλόμενα και πιο εύκολα επηρεαζόμενα από το precision κάθε αλγορίθμου και την ύπαρξη θορύβου.



### 4.3.2 ΟΡΙΣΜΟΣ ΤΩΝ TESTS

Έχω λοιπόν 5 μετρικές συν επιπλέον την προσωπική μας επισκόπηση της ποιότητας του αποτελέσματος. Σε αυτά θα πρέπει να προστεθούν και έλεγχοι άλλης μορφής, όπως η χρονική απόδοση των αλγορίθμων (ποιος είναι πιο γρήγορος) και «θεωρητική» επισκόπηση τους, ποιος επηρεάζεται περισσότερο από εξωτερικούς παράγοντες όπως φωτισμός και ποιος εξαρτάται περισσότερο από ευμετάβλητα δεδομένα (πχ ο Poisson επηρεάζεται από το μέγεθος παραθύρου, ο Gaussian από το precision του noise estimation κτλ). Επίσης πρέπει να δούμε πως αντιμετωπίζουν τις ιδιάζουσες περιπτώσεις που συναντήσαμε στα video αυτά, δηλ. το flickering στο 1<sup>ο</sup> video και το freezing στο 2<sup>ο</sup>.

Επιπλέον, δεν είναι δυνατόν να παρουσιάσουμε εδώ όλες τις μετρικές για όλες τις εικόνες. Επιλέγουμε λοιπόν κάποια χαρακτηριστικά στιγμιότυπα από τα αποτελέσματα (που οπτικά βλέπουμε πως ακολουθούν τη γενική συμπεριφορά κάθε αλγόριθμου). Έτσι παρακάτω, έχω συνολικά 36 κριτικές ( 18 για κάθε video σε μια χαρακτηριστική change mask image).

Τέλος, σε πολλούς αλγορίθμους έχω το φαινόμενο της παρουσίας θορύβου, δηλ. ακανόνιστων changed pixels σε διάφορα σημεία της change mask. Κάποιοι θα μπορούσαν να πουν ότι ο θόρυβος αυτός δεν είναι σημαντικός για την απόδοση του αλγόριθμου, μπορεί να αφαιρεθεί με κάποιο τρόπο. Σωστό, γι' αυτό και μεις **επιλέγουμε να υπολογίσουμε τις μετρικές ελέγχου στις περιοχές των πραγματικών αλλαγών και όχι στη συνολική εικόνα (ορίζουμε παράθυρα γύρω από αυτές)**. Με αυτό τον τρόπο ελέγχω μόνο την επίδραση του θορύβου στις μορφοποίηση των regions των πραγματικών αλλαγών ( ούτως ή άλλως, η παρουσία του θορύβου γενικά στη change mask θα ειδωθεί εμπειρικά από εμάς) και επιπλέον, βελτιώνω τη ταχύτητα των tests. Επιπρόσθετα, έτσι μπορώ να απλοποιήσω τον υπολογισμό του compactness, αντί να βρίσκω κάθε region ( labeling ) και σε αυτή ποια pixels της είναι όρια της region, τώρα μπορώ να απλοποιήσω τους υπολογισμούς. Θέτω λοιπόν μια «μέση» compactness (που και διαισθητικά προσεγγίζει το αποτέλεσμα που θα είχα με τον προηγούμενο τρόπο)

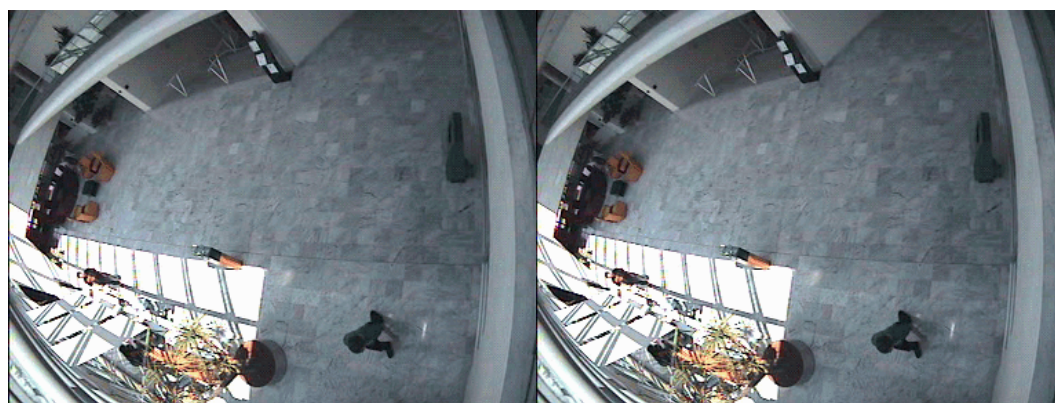
## Για το παράθυρο που έχω πραγματική αλλαγή

**Area = approximately all non-zero pixels**

**Perimeter= approximately all non-zero pixels with at least one zero 8-neighbor pixel**

### 4.3.3 ΑΠΟΤΕΛΕΣΜΑΤΑ

#### 4.3.3.1 TEST 1

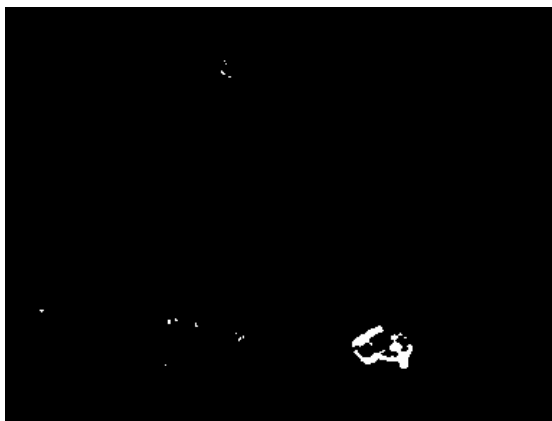
**IN-1****IN**



**1) EULER ΣE 256 COLOR GRAYSCALE**



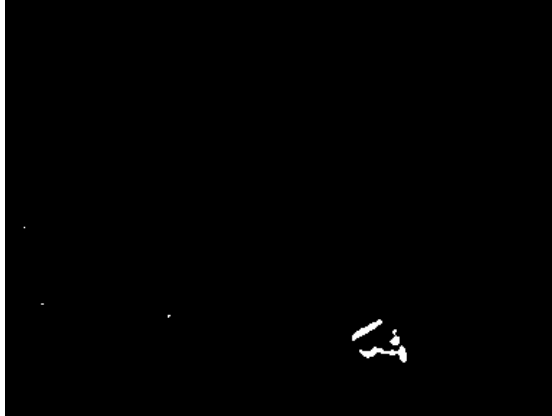
**2) EULER ΣE 256 COLOR GRAYSCALE ME MEDIAN ΣTO DIFFERENCE**



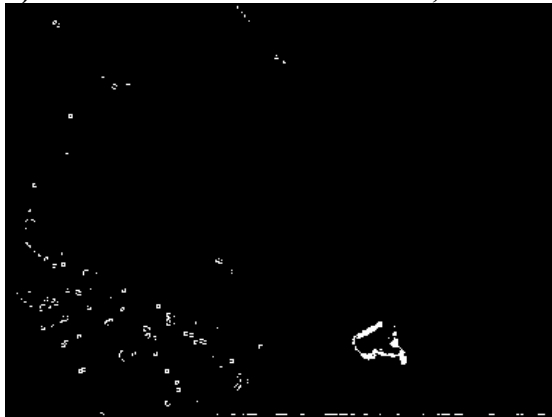
**3) EULER ME 24 BIT-COLOR IMAGES**



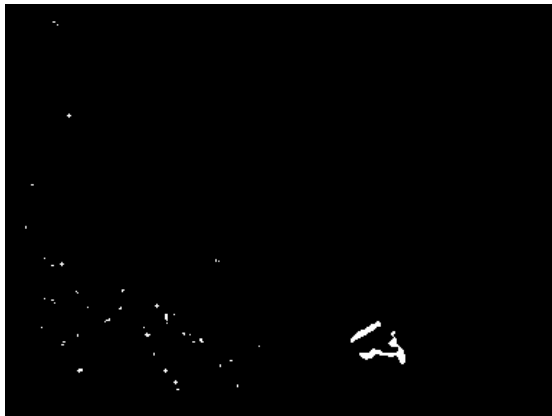
**4) EULER ΣΕ 24 BIT-COLOR IMAGES ME MEDIAN ΣΤΟ DIFFERENCE**



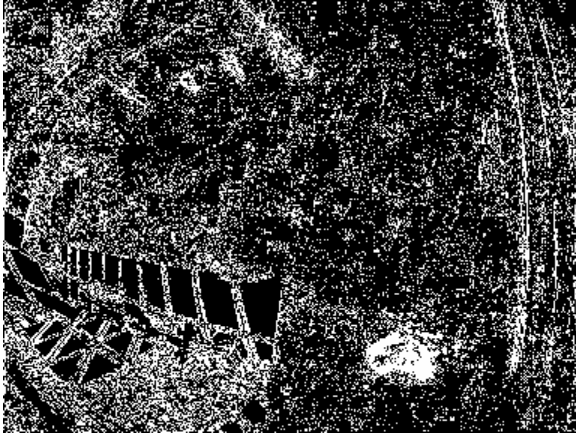
**5) EULER ΣΕ 24 BIT-COLOR, HOMOMORPHIC FILTERED IMAGES**



**6) EULER ΣΕ 24 BIT-COLOR, HOMOMORPHIC FILTERED IMAGES ME MEDIAN FILTERED DIFFERENCE IMAGE**



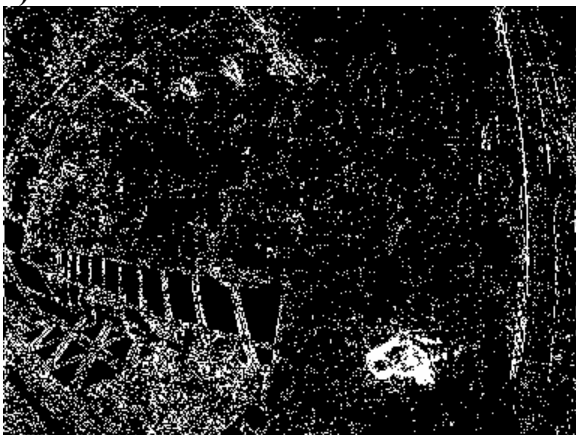
**7) GAUSSIAN ΣE 256 COLOR GRAYSCALE**



**8) GAUSSIAN – 256 COLOR GRAYSCALE – MEDIAN ΣTO DIFFERENCE**



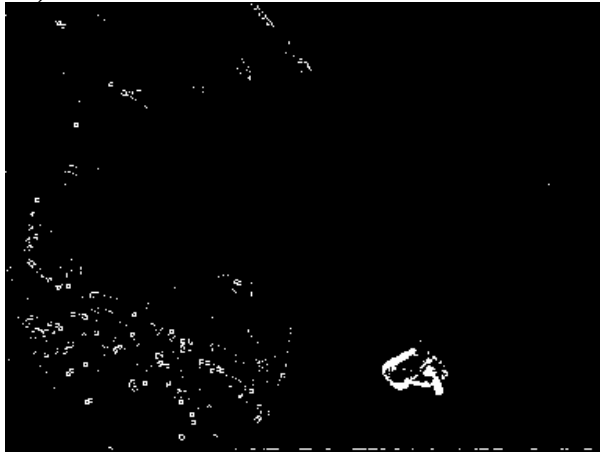
**9) GAUSSIAN- 24BIT COLOR**



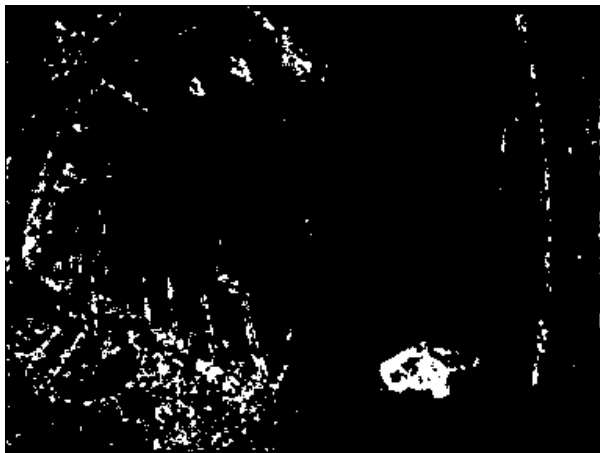
**10) GAUSSIAN- 24BIT COLOR- MEDIAN FILTER  $\Sigma$ TO DIFFERENCE**



**11) GAUSSIAN- 24BIT COLOR- HOMOMORPHIC PREFILTERING**



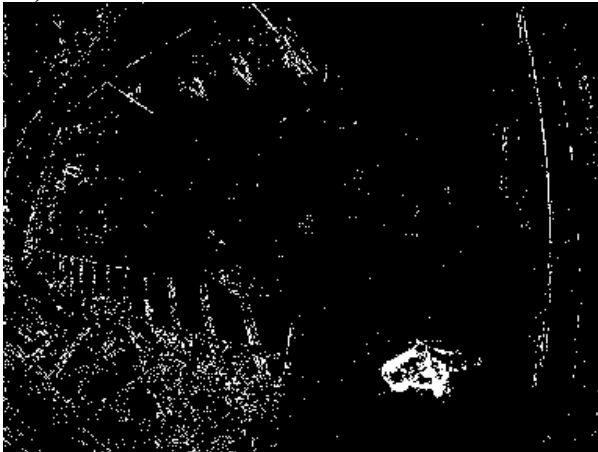
**12) GAUSSIAN- 24BIT COLOR-HOMOMORPHIC PREFILTERING-MEDIAN  $\Sigma$ TO DIFFERENCE**



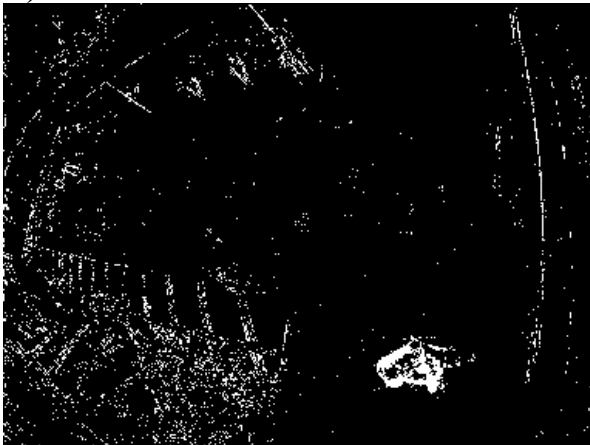
**13) POISSON- 256 COLOR GRAYSCALE**



**14) POISSON- 256 COLOR GRAYSCALE- MEDIAN  $\Sigma$ TO DIFFERENCE**



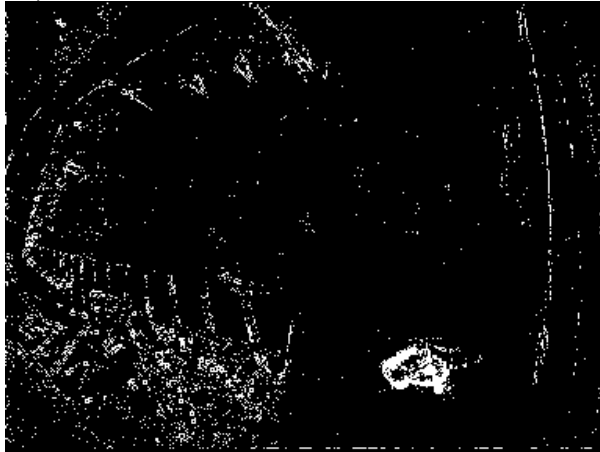
**15)POISSON-24BIT COLOR**



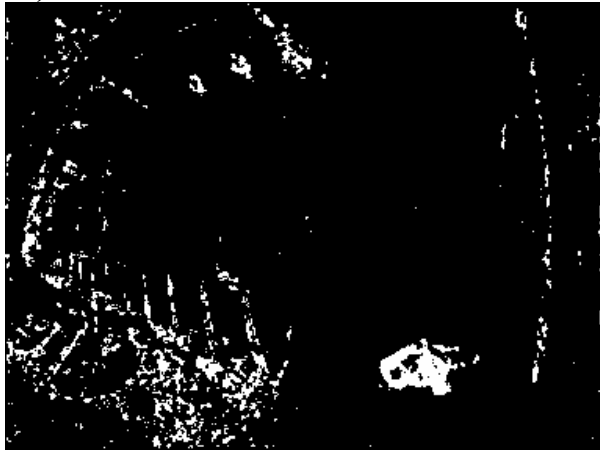
**16) POISSON- 24BIT COLOR- MEDIAN  $\Sigma$ TO DIFFERENCE**



**17) POISSON- 24BIT COLOR- HOMOMORPHIC PREFILTERING**



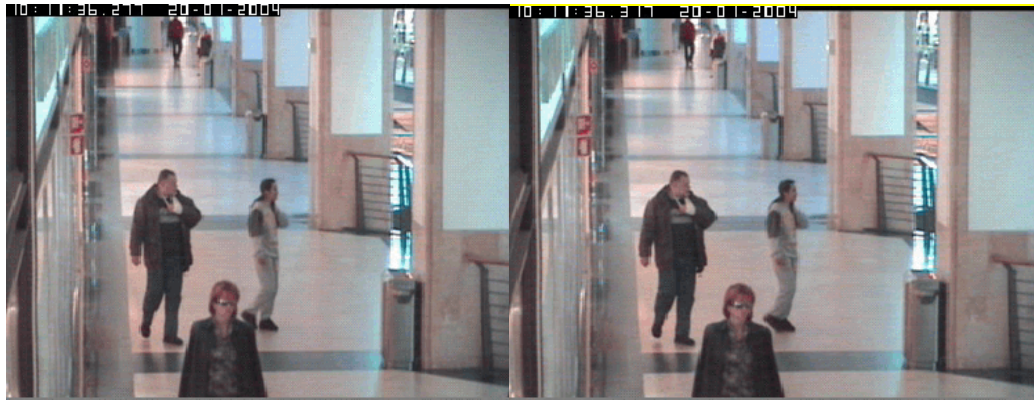
**18) POISSON- 24BIT COLOR- HOMOMORPHIC FILTERING- MEDIAN**



#### 4.3.3.2 TEST 2

**In-1**

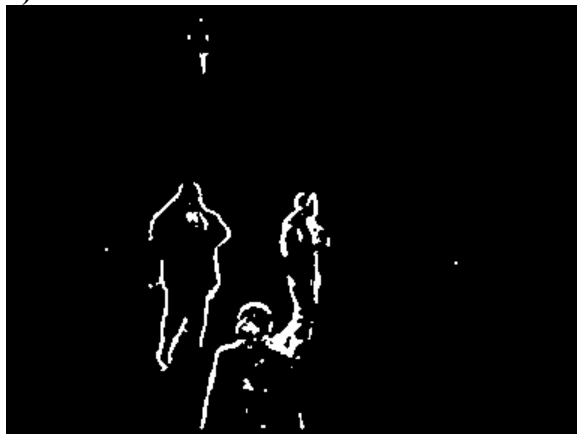
**In**



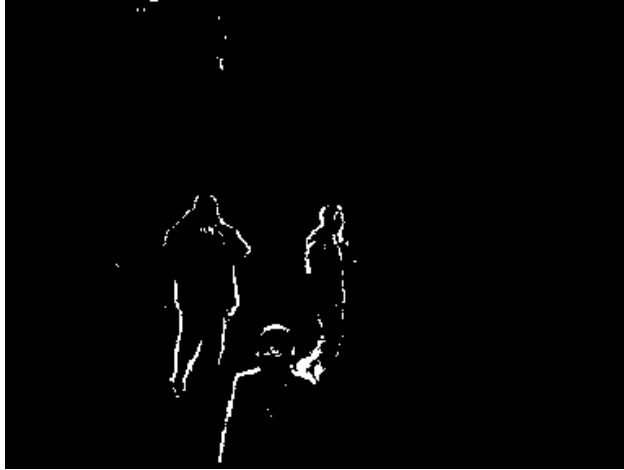
##### 1) EULER 256 COLOR GRAYSCALE



##### 2) EULER- 256 COLOR GRAYSCALE- MEDIAN ΣTO DIFFERENCE



### 3) EULER- 24BIT COLOR



### 4) EULER- 24BIT COLOR- MEDIAN $\Sigma$ TO DIFFERENCE

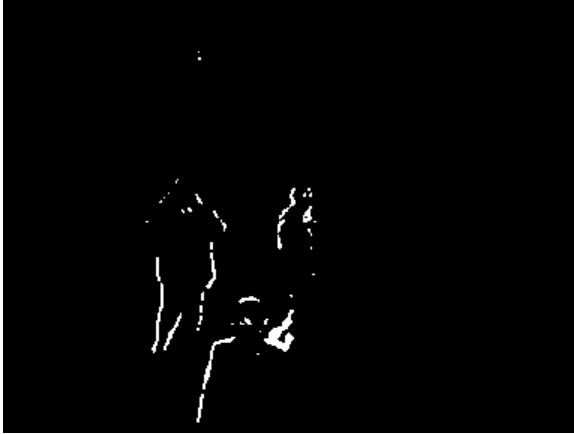


### 5) EULER- 24 BIT COLOR- HOMOMORPHIC PREFILTERING

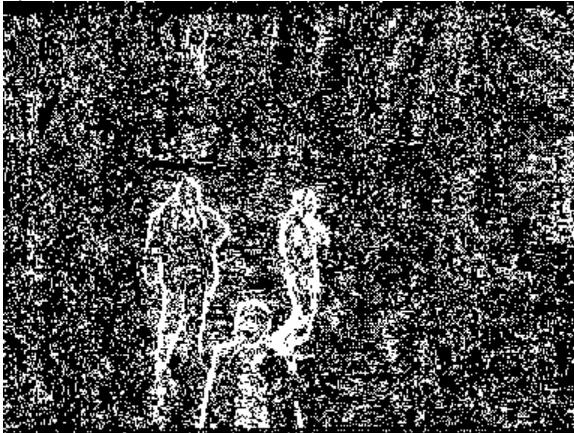




**6) EULER-256 COLOR- HOMOMORPHIC PREFILTERING-MEDIAN**



**7) GAUSSIAN- 256 COLOR GRAYSCALE**



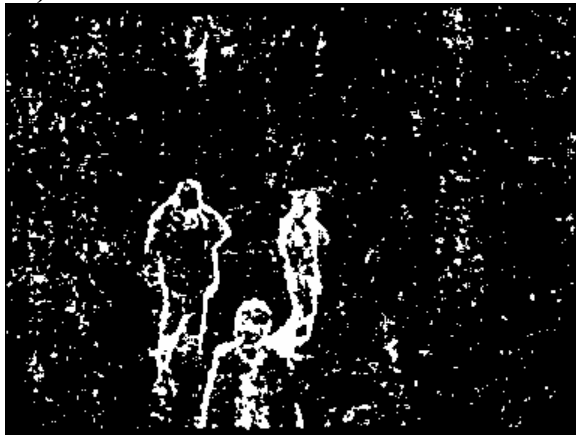
**8) GAUSSIAN- 256 COLOR GRAYSCALE- MEDIAN  $\Sigma$ TO DIFFERENCE**



**9) GAUSSIAN- 24 BIT COLOR**



**10) GAUSSIAN- 24BIT COLOR- MEDIAN  $\Sigma$ TO DIFFERENCE**



**11) GAUSSIAN- 24 BIT COLOR- HOMOMORPHIC PREFILTERING**



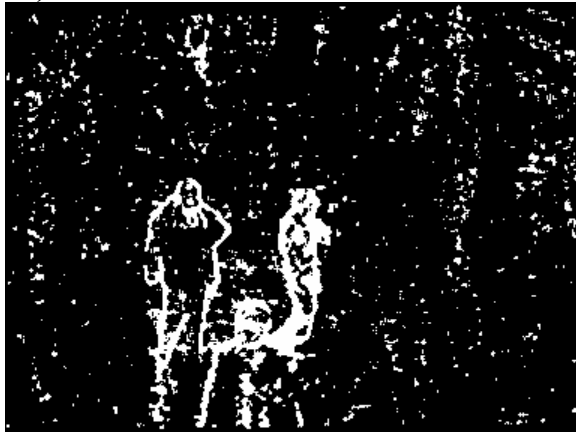
**12) GAUSSIAN- 24 BIT COLOR- HOMOMORPHIC- MEDIAN**



**13) POISSON- 256 COLOR GRAYSCALE**



**14) POISSON- 256 COLOR GRAYSCALE- MEDIAN**



**15) POISSON- 24 BIT COLOR**



**16) POISSON- 24 BIT COLOR- MEDIAN**



**17)POISSON – 24BIT COLOR- HOMOMORPHIC PREFILTERING**



## 18) POISSON- 24BIT COLOR- HOMOMORPHIC- MEDIAN



Και ο υπολογισμός των μετρικών σύγκρισης

### 1)TEST 1

**ΣΕ                      X                      V                      Eul                      Comp/ness                      Labels**

#### i) EULER

<b>GR</b>	38	25	13	180	13
<b>GR/MED</b>	32	28	4	157	5
<b>COL</b>	36	24	12	178	12
<b>COL/MED</b>	26	23	3	143	4
<b>COL/HOM</b>	28	23	5	170	9
<b>COL/HM/MD</b>	24	22	2	146	4

#### ii)GAUSSIAN

<b>GR</b>	424	166	258	1371	314
<b>GR/MED</b>	106	62	44	341	74
<b>COL</b>	297	90	207	822	235
<b>COL/MED</b>	67	51	16	240	37
<b>COL/HOM</b>	43	30	13	217	14
<b>COL/HM/MD</b>	41	41	0	202	11

### iii) POISSON

<b>GR</b>	199	72	127	597	163
<b>GR/MED</b>	84	52	32	296	55
<b>COL</b>	74	47	27	380	40
<b>COL/MED</b>	42	47	-5	225	13
<b>COL/HOM</b>	83	56	27	390	43
<b>COL/HM/MD</b>	41	44	-3	211	12

## 2) TEST 2

### i) EULER

<b>ΣE</b>	<b>X</b>	<b>V</b>	<b>Eul</b>	<b>Comp/ness</b>	<b>Labels</b>
<b>GR</b>	233	130	103	1505	114
<b>GR/MED</b>	184	152	32	1412	55
<b>COL</b>	165	84	81	894	86
<b>COL/MED</b>	129	87	42	887	53
<b>COL/HOM</b>	155	73	82	767	77
<b>COL/HM/MD</b>	114	72	42	729	47

### ii) GAUSSIAN

<b>GR</b>	1809	863	946	8374	600+
<b>GR/MED</b>	631	440	191	3230	312
<b>COL</b>	1436	409	1027	4773	600+
<b>COL/MED</b>	475	302	173	2324	244
<b>COL/HOM</b>	724	251	473	2776	502
<b>COL/HM/MD</b>	282	216	66	1737	105

### iii) POISSON

<b>GR</b>	1366	400	966	4623	600+
<b>GR/MED</b>	469	317	152	2391	224
<b>COL</b>	599	225	374	2430	383
<b>COL/MED</b>	261	206	55	1681	97
<b>COL/HOM</b>	401	186	215	1941	235
<b>COL/HM/MD</b>	304	219	85	1743	117

### 4.3.4 ΑΝΑΣΚΟΠΗΣΗ ΔΥΣΚΟΛΩΝ ΠΕΡΙΠΤΩΣΕΩΝ

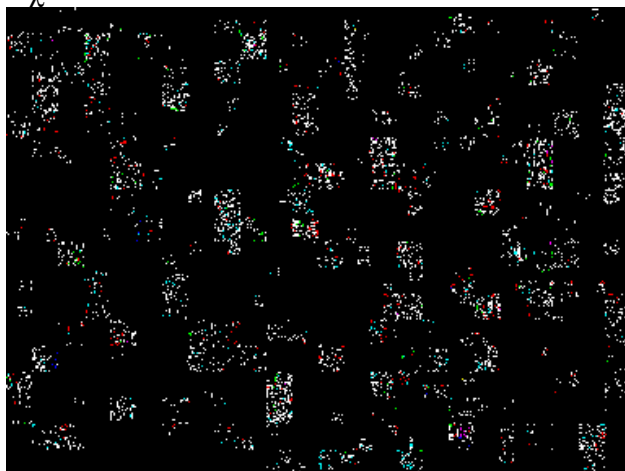
#### 1) FLICKERING ΣΤΟ 1o VIDEO

Το φαινόμενο παρουσιάζεται σε όλο το video, δεν έχει σταθερή περίοδο εμφάνισης, και είναι αρκετά συχνό. Το αποτέλεσμα του, ειδικά στην επιλογή Iv - Iv-1 που επιλέξαμε εδώ, είναι «διόγκωση» των regions στην change mask και μετακίνηση τους στο χώρο. Ο καλός λοιπόν αλγόριθμος εδώ ξεχωρίζει από το ότι δίνει μικρές σε πάχος regions, που εδώ είναι «γραμμές». Πάλι καλύτερος εδώ ο euler με poisson/Gaussian να αποδίδουν παρόμοια μεταξύ τους. Βλέπουμε λοιπόν, ότι αν και στη θεωρία οι αλγόριθμοι αυτοί κρίνονται εντελώς ακατάλληλοι λόγω της spatial κίνησης της κάμερας, μπορούν να αποδώσουν σχετικά καλά και ειδικά ο euler αποδίδει αρκετά καλά, για τη μικρή αυτή μεταβολή της θέσης της.

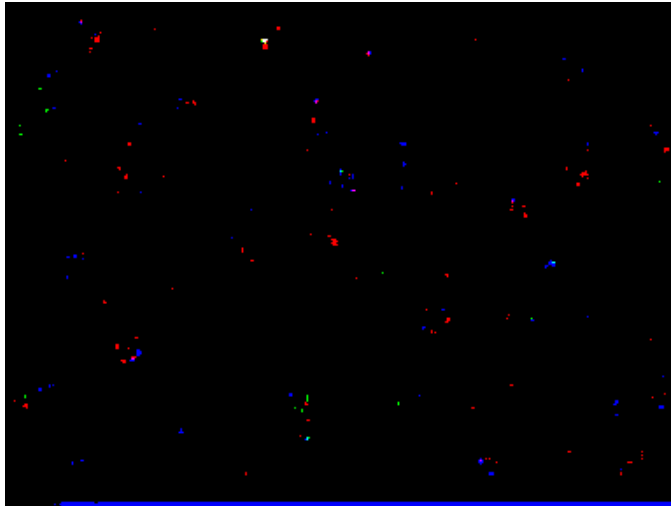
#### 2) FREEZING ΣΤΟ 2o VIDEO

Το φαινόμενο, με μια εμφάνιση περίπου ανά 4-7 frames, έχει ως αποτέλεσμα την παρουσία change mask χωρίς σημαντική πληροφορία, εννοώντας ότι δεν έχουν πραγματικές regions αλλά μόνο θόρυβο, αν οι αλγόριθμοι ήταν βέλτιστοι οι change masks θα ήταν κενές εντελώς.

Πχ



**A. CHANGE MASK ΜΕΤΑΞΥ FRAMES ΠΟΥ ΕΜΦΑΝΙΣΤΗΚΕ FREEZING (GAUSSIAN ΣΕ GRAYSCALE)**



#### **B. CHANGE MASK ΜΕΤΑΞΥ ΤΩΝ ΙΔΙΩΝ FRAMES ΠΟΥ ΕΜΦΑΝΙΣΤΗΚΕ FREEZING (GAUSSIAN ΣΕ COLOR/HOMOMORPHIC).**

Γενικά, σε όλους τους αλγόριθμους, τα artifacts αυτά είναι πολλά στο change detection με εικόνες grayscale. Μειώνονται όμως δραστικά με το πέρασμα μας σε 24 bit color εικόνες και με χρήση homomorphic σχεδόν εξαλείφονται (η συντριπτική τους πλειοψηφία εμφανίζει ένταση σε 1 ή 2 κανάλια χρώματος και άρα απαλείφονται με χρήση AND FUSION).

#### **4.3.5 ΣΥΜΠΕΡΑΣΜΑΤΑ**

##### **1) ΣΗΜΑΙΝΩΝ ΡΟΛΟΣ ΤΗΣ ΠΡΟΕΠΕΞΕΡΓΑΣΙΑΣ (FEATURE ANALYSIS)**

Τα αποτελέσματα αποδεικνύουν ότι οι σωστές επιλογές εδώ βελτιώνουν σημαντικά την απόδοση των αλγορίθμων. Για άλλη μια φορά λοιπόν επαναλαμβάνουμε, ένας καλός αλγόριθμος change detection δεν είναι μόνο το classification αλλά όλο το σύνολο των διαδικασιών του ( Feature Extraction/Analysis, Classification κτλ, πράγμα που παρεξηγούν πολλοί)

Η απόδοση μας βελτιώνεται σημαντικά διαδοχικά με τη μετάβαση από ασπρόμαυρο σε έγχρωμο και τελικά την προσθήκη homomorphic filtering. Οι regions μειώνονται, προσεγγίζοντας τελικά τον πραγματικό αριθμό τους, διαχωρίζονται και η δομή τους τείνει σε αυτή που θέλουμε.

Είναι χαρακτηριστικό ότι ο Gaussian κυριολεκτικά «απογειώνει» την απόδοση του με τη χρήση homomorphic. Για το τελευταίο βέβαια, πρόβλημα αποτελεί η εμφάνιση noise artifacts ( μετακινούμενα «κουτάκια») στο video 1, στην περιοχή του παραθύρου, εκεί δηλαδή που η επίδραση του εξωτερικού φωτός είναι μεγάλη. Αυτό εξηγείται από τις σημαντικές διακυμάνσεις που έχει η φωτεινότητα μεταξύ γειτονικών περιοχών (το φως «κτυπά» κάθετα κάποια pixels ) με αποτέλεσμα πολύ μεγάλη τιμή illumination τοπικά δηλ. δεν μπορώ να διακρίνω το reflectance. Τα artifacts αυτά μετακινούνται (το φως του ήλιου αλλάζει γωνία πρόσπτωσης ανά εικόνα, με αποτέλεσμα να φαίνονται ως πραγματικές αλλαγές. Είναι όμως μικρή η επίδραση τους στην πραγματική πληροφορία και σχεδόν εξαλείφονται με χρήση του median filtering στο preprocessing. Δεν αποτελούν λόγο μη χρήσης του homomorphic αφού είναι αυτό που δρα ως η «καθαριότητα» δύναμη για πλήθος άλλων προβλημάτων στις εικόνες. Είναι αυτό που μας διακρίνει ότι οι σκιές των αντικειμένων γενικά τείνουν να έχουν ένταση σε μια ή δύο χρωματικές μεταβλητές (R, G ή B) και άρα να αφαιρεθούν με απλή χρήση AND FUSION, είναι αυτό που



εξομαλύνει της μεταβολές φωτεινότητας σε μια region ( και άρα μειώνει το φαινόμενο των “σπασμένων” regions στο αποτέλεσμα) και πλήθος άλλα.

## **2) ΒΕΛΤΙΣΤΗ ΑΠΟΔΟΣΗ ΤΟΥ EULER ALGORITHM**

Ο αλγόριθμος με χρήση Euler number Descriptor είναι γενικά αυτός που αποδίδει καλύτερα, τόσο σε επίπεδο οπτικό (όπου τα changes ευρίσκονται με αρκετά μεγάλη ακρίβεια) όσο και σε επίπεδο των μετρικών που θέσαμε, με των αριθμό των labels να προσεγγίζει τον Euler number (δηλ. η δομή των region βρέθηκε με αρκετή πιστότητα). Οι Gaussian/Poisson παρουσιάζουν γενικά όμοια απόδοση αλλά σίγουρα κατώτερη από αυτή του Euler. Σε αυτό συντείνει και το γεγονός ότι ο Euler είναι πρακτικά ανεξάρτητος από μεταβλητές, δίνει για μια συγκεκριμένη εικόνα ένα πάντα αποτέλεσμα και δεν εξαρτάται όπως ο Poisson από το μέγεθος παραθύρου ή όπως ο Gaussian από την ακρίβεια του noise estimation και του ποσοστού ανοχής  $P_n$ .

Γενικά, για ένα συγκεκριμένο set προεπεξεργασίας, ο Gaussian δίνει το υψηλότερο κατώφλι, ο Poisson μικρότερο ενώ ο Euler δεν μπορεί να τεθεί κάπου ανάμεσα τους, ως αλγόριθμος με διαφορετική λογική ( περιγραφικός της πληροφορίας και όχι του θορύβου ). Το κατώφλι εδώ πότε είναι μικρότερο πότε μεγαλύτερο, πότε ανάμεσα τους αλλά γενικά είναι το βέλτιστο.

Βέβαια, ο Euler είναι και ο πιο χρονοβόρος αλγόριθμος, περίπου 60 sec/frame ( Gaussian 10 sec/frame, Poisson 30 sec/frame). Οι χρονικές απαιτήσεις των μη-classification τμημάτων του συστήματος (homomorphic κτλ) είναι πολύ μικρές ( όχι πάνω συνολικά από 2-4 sec/frame).

## **3) ΧΡΗΣΗ MEDIAN FILTERING ΣΤΗΝ CHANGE MASK**

Το median filtering δρα γενικά ως smoothing πάνω στο αποτέλεσμα, σβήνει τα απομονωμένα pixels (που έχουν δημιουργηθεί λόγω θορύβου κτλ) στις change mask και «εντονοποιεί» τις μεγαλύτερες regions (που μπορεί κάποιες από αυτές να μην είναι πραγματικές αλλαγές). Άρα έχει και καλό και κακό αντίκτυπο.

Ουσιαστική βελτίωση δίνει μόνο στον Euler αλγόριθμο που έχω κυρίως μικρές περιοχές θορύβου, αλλά και πάλι μεταβάλλει τη μορφολογία και των πραγματικών change regions, πράγμα που θα συζητήσουμε αν το κάνει χρήσιμο στο tracking παρακάτω.

## **4) AND FUSION**

Το and Fusion γενικά αποδείχθηκε καλή επιλογή ως τρόπος συνδυασμού των αποτελεσμάτων στα 3 κανάλια χρώματος. Αν και, όπως δείξαμε στην εισαγωγή, μπορεί να χαθεί έτσι και σημαντική πληροφορία (που εμφανίζεται μόνο σε 1 ή 2 κανάλια χρώματος), καταφέρνει να δώσει (ειδικά στον Euler) σχεδόν όλες τις πραγματικές αλλαγές και ελάχιστο θόρυβο, που μειώνεται όσο προχωρώ σε color και homomorphic preprocessing. Άρα τελικά, ήταν και η τελική μας επιλογή για συνδυασμό των R G B αποτελεσμάτων, διότι είναι εύκολο στην υλοποίηση, γρήγορο στην εφαρμογή και αποδίδει καλά.

## **ΚΕΦΑΛΑΙΟ 5: ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ OBJECT TRACKING**

- Τελική Επιλογή Βέλτιστης Διαδικασίας Change Detection
- Υλοποίηση Kalman Filtering, Αποτελέσματα Και Συμπεράσματα

## 5. ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ OBJECT TRACKING

### 5.1 ΤΕΛΙΚΗ ΕΠΙΛΟΓΗ ΜΕΘΟΔΟΛΟΓΙΑΣ CHANGE DETECTION

Παίρνοντας τη σκυτάλη από το change detection τμήμα της υλοποίησης μας έχουμε καταλήξει σε ένα σύστημα change detection που πιστεύουμε βάσιμα πως μπορεί να δώσει καλά αποτελέσματα, και στα δύο videos που έχουμε.

24bit Color Sequencies ==>> Homomorphic Filtering ==>> Euler Number Algorithm

Ο σκοπός μας τώρα είναι μέσω change detection να απομονώσουμε τα moving objects από το background σε κάθε frame ώστε να έχουμε μια κατάλληλη μορφή των frames για εφαρμογή tracking. Έτσι, κατ' αρχάς παράγουμε ένα background για κάθε video. Ο τρόπος εύρεσης που επιλέγουμε είναι η εφαρμογή μέσης τιμής σε κάθε θέση pixel σε όλα τα frames του ίδιου video.

Έτσι, μέσω αυτής της μεθόδου μέσης τιμής προέκυψαν τα backgrounds

#### 1) VIDEO 1



#### 2) VIDEO 2

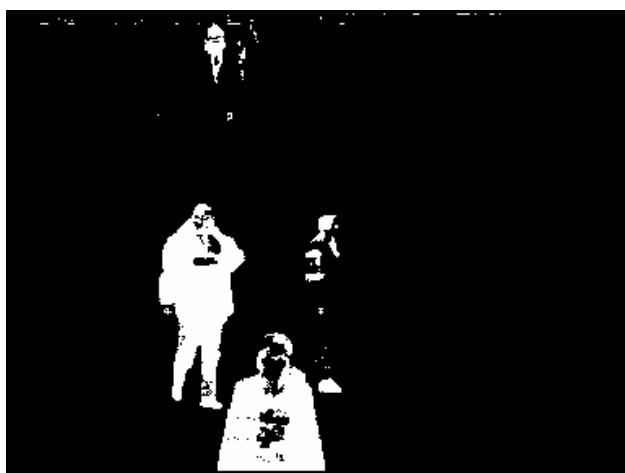
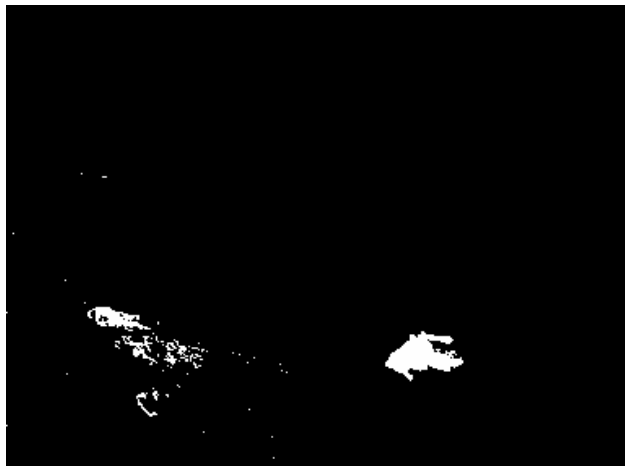


Παρατηρούμε ότι με τον απλό αυτό τρόπο έχουμε αρκετά καλή προσέγγιση και άρα δεν χρειάστηκε να καταφύγουμε σε κάποια πιο sophisticated μέθοδο. Χαρακτηριστικό είναι πως στο 2<sup>ο</sup> video δεν υπήρχε frame χωρίς κίνηση, αλλά η μέθοδος αυτή μέσης τιμής έδωσε την εικόνα που δεν έχει κίνηση = background.

Το παραπάνω έγινε για να κάνουμε change detection στα δυο videos. Επιλέγουμε το difference του FRAME – BACKGROUND για να δημιουργήσουμε το αρχικό workload στο change detection εδώ. Έτσι, αντί απ' ευθείας της χρήσης των αποτελεσμάτων του κεφ. 4 δεν θα χρειαστεί να κάνουμε dilations, erosions κτλ ώστε να έχουμε «γεμισμένα» αντικείμενα, ενώ προβλήματα πχ το freezing στο 2<sup>ο</sup> video θα επηρεάσει βεβαία παρακάτω το tracking **αλλά** δεν θα έχω προβλήματα όπως πχ το αντικείμενο εξαφανίζεται- επανεμφανίζεται κτλ λόγω του freezing.

Σε πρώτο στάδιο λοιπόν κάνω change detection ( 24 bit color- Homomorphic filtering – Euler) στα image differences των frames του video με τα στατικά backgrounds που υπολογίσαμε παραπάνω. Αποφεύγω το median filtering στα αποτελέσματα για κάποιο λόγο που θα εξηγήσουμε αμέσως μετά.

Τελικά τα αποτελέσματα μας δικαιώνουν, για την ακρίβεια απλώς δικαιώνουν τις μετρήσεις μας στο κεφ 4



Βεβαία, δεν είναι τέλεια, πχ η κοπέλα στα δεξιά του άντρα στο 2<sup>ο</sup> video, **συνεχώς** γίνεται detected με έλλειψη τμημάτων του object που την αποτελεί, είτε με homomorphic είτε όχι. Έχουμε μια περίπτωση «καμουφλάζ» που λύνεται με χρήση

υπέρυθρου φωτισμού κτλ. Δεν ασχολιόμαστε ιδιαίτερα, τα υπόλοιπα changes βρίσκονται αρκετά καλά.

## 5.2 ΜΕΤΑΒΑΣΗ ΣΕ TRACKING- ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ

### 5.2.1 ΟΡΙΣΜΟΣ ΤΟΥ WORKLOAD ΓΙΑ ΤΟ TRACKING

Λίγο πριν, είπαμε πως αποφύγαμε το median filtering στο change detection. Ο λόγος είναι ο εξής, σε αυτό ακριβώς το στάδιο θέλω να έχω μια κατάλληλη μορφή εικόνας ώστε να μπορέσω κάθε αντικείμενο να είναι όσο το δυνατόν ξεχωριστό και καλά ορισμένο στο frame, αν έκανα λοιπόν median ρισκάριζα, πχ να έχω «σύνδεση» σε κάποια σημεία μεταξύ objects, γενικά πείραγμα της δομής των αντικειμένων.

Ο λόγος που θέλω όσο το δυνατόν καλά ξεχωρισμένα αντικείμενα είναι η μορφή που ορίζω ως είσοδο στον tracking αλγόριθμο. Αποφασίζω λοιπόν, για μια σειρά εικόνων (πχ 150-200 σε κάθε video) να κάνω tracking για ένα αντικείμενο, συγκεκριμένα το κέντρο βάρους του. Έτσι, πριν περάσω στο κύριο αλγόριθμο tracking κάνω τα εξής:

1) Πρώτα κάνω labeling στο αποτέλεσμα του change detection, connected component labeling, όπως ακριβώς το όρισα στη θεωρία και έκανα χρήση αυτού στα test του κεφ. 4.

Το αποτέλεσμα, παρουσιάζει ατέλειες, θα πρέπει να πούμε ότι αυτές οφείλονται τόσο στα αποτελέσματα του change detection όσο και στο ότι ο ίδιος ο αλγόριθμος δεν είναι βέλτιστος [12].

2) Στα αποτελέσματα αυτά, θέλω να κάνω tracking. Επιλέγω λοιπόν να παρακολουθώ το κέντρο βάρους ενός object, έστω για κάποιο αριθμό frames.

Εδώ λοιπόν, διατρέχω τα labeled αποτελέσματα και επιλέγω το αντικείμενο σε κάθε frame που θέλω να κάνω tracking. Το πρόγραμμα βρίσκει έπειτα το κέντρο βάρους του αντικειμένου και το μαρκάρει, αν πχ κάνω επιλογή σε θέση που έχει ένταση K τότε για κάθε pixel με ένταση K στο frame Τότε, το κέντρο βάρους

## Area – Center

- Binary (or gray) region  $B[i,j]$   
–  $B[i,j] = 1$  if  $(i,j)$  in the region, 0 otherwise

– Area: 
$$A = \sum_{i=1}^N \sum_{j=1}^M B[i,j]$$

– Center of gravity: 
$$x_0 = \frac{\sum_{i=1}^N \sum_{j=1}^M jB[i,j]}{A} \quad y_0 = \frac{\sum_{i=1}^N \sum_{j=1}^M iB[i,j]}{A}$$

### ΕΙΚΟΝΑ 5.2.1 Object Area Definition [12]

Όπου αντί 1 έχω K.

Δηλ. το σύστημα που θέλω να προβλέπω την κίνηση του είναι ένα σημείο, το κέντρο βάρους του αντικειμένου. Αν είχαμε real time εφαρμογή όλα αυτά θα γινόντουσαν σε κάθε frame ξεχωριστά, δηλ

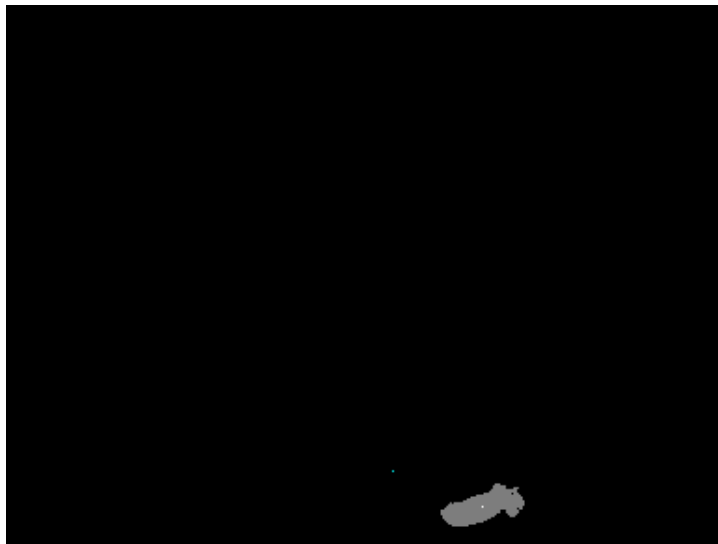
Change Detection στο frame N → Tracking και πρόβλεψη για το N+1 → Change Detection στο N+1 ... κτλ.

Έτσι ας δούμε μερικές από τις εικόνες εισόδου που θα έχω στο tracking και το κέντρο βάρους που βρήκαμε.

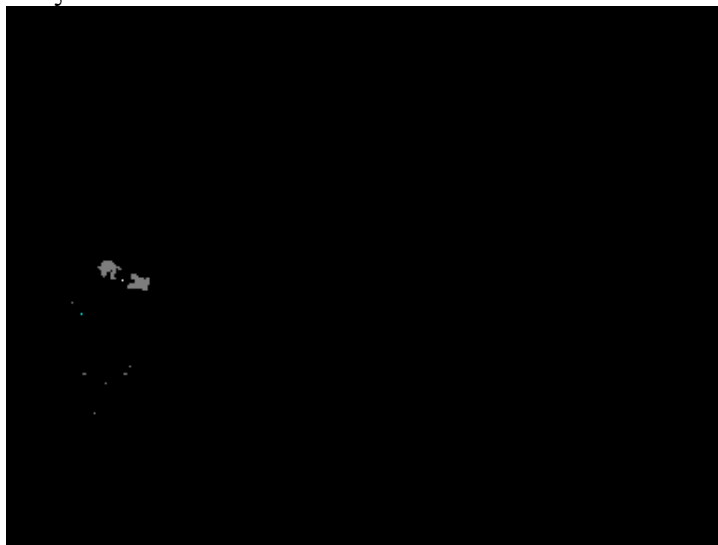
### 1) VIDEO 1

Αποφασίζω να παρακολουθώ την κίνηση ενός συγκεκριμένου ανθρώπου για περίπου 200 frames

Από frame 77



Έως frame 332



Το άσπρο σημάδι ορίζει το κέντρο βάρους του αντικειμένου που θα παρακολουθώ,

Δηλ. θέλω να βρίσκω τον άνθρωπο που ξεκινά από κάτω δεξιά και αφού χαιρετηθεί με έναν άλλο στο κέντρο καταλήγει στα αριστερά της εικόνας



Μέσω της παρακολούθησης του κέντρου βάρους.

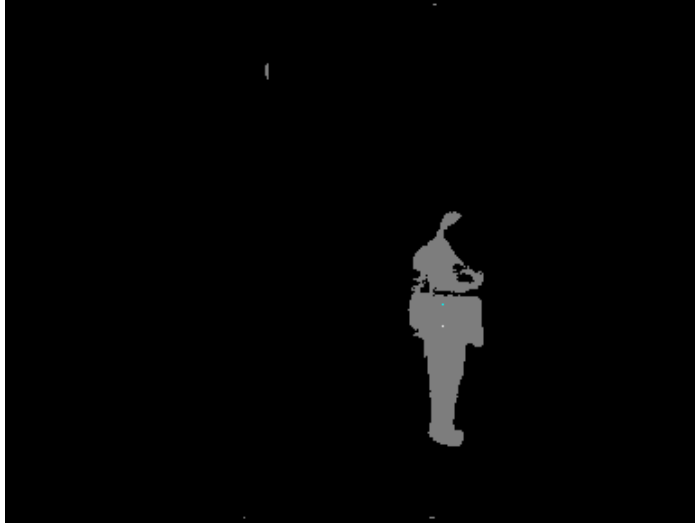
## 2) VIDEO 2

Ανάλογα παρακολουθώ τον κύριο στο κέντρο του video 2

Από frame 2



Έως frame 123



δηλ. το αντικείμενο



Έτσι, πλέον το tracking ανάγεται σε εύρεση σε κάθε βήμα μιας πρόβλεψης στο Frame N για το κέντρο βάρους που έχω στο frame N+1, μέσω του kalman filter

## 5.2.2 ΑΝΑΠΤΥΞΗ WORKLOAD

Όπως είπαμε, χρειαζόμαστε σε κάθε βήμα μια μέτρηση  $y_n$  της πραγματικής κατάστασης ώστε να διορθώνουμε τη πρόβλεψη μας από  $x_{n,n-1}^*$  σε  $x_{n,n}^*$ . Μέσω του change detection έχουμε αρκετά καλά ξεχωρίσει τα αντικείμενα μεταξύ τους και συνεπώς αυτή η επιλογή μπορεί εύκολα να γίνει.

Θα μπορούσαμε απλώς τώρα σε κάθε frame να επιλέγουμε σημείο πρόβλεψης **ακριβώς** το πραγματικό κέντρο, τότε αφού η κίνηση των αντικειμένων που έχω είναι **γραμμική, τα αποτελέσματα θα άγγιζαν το τέλειο**. Ωστόσο, το μέλημα μας είναι να ελέγξουμε κάπως πιο αυστηρά το φίλτρο μας (από ότι επιβάλλει η θεωρία), με

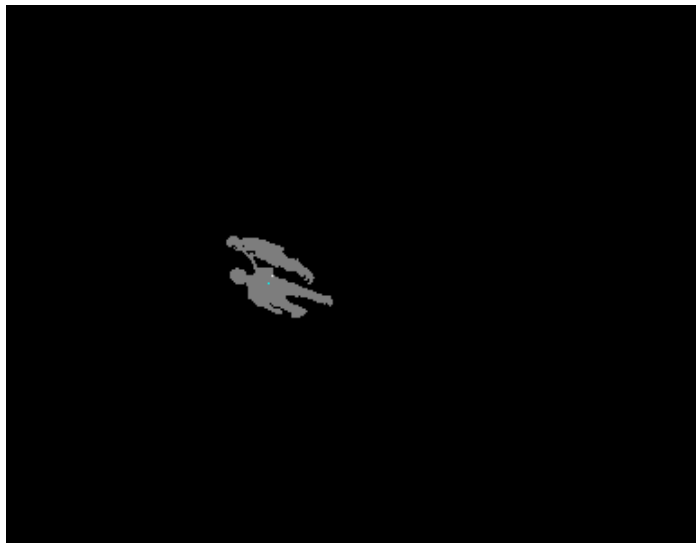


ανάλογο τρόπο που δείξαμε πως σε περιορισμένη κίνηση κάμερας οι αλγόριθμοι change detection που επιλέξαμε καταφέρνουν ν' αντεπεξέλθουν. Θέλουμε λοιπόν να δούμε τη συμπεριφορά του σε μια ,έστω, περιορισμένη μη γραμμική κίνηση.

Γι' αυτό το λόγο, επεξεργαζόμαστε περαιτέρω το workload που έχουμε από το change detection ως εξής:

- Το connected component labeling που επιλέγουμε είναι από τη φύση του μη τέλει , αλλά και εδώ επιβαρύνεται το αποτέλεσμα του από αντικείμενα που «συγχέονται» κτλ, άρα οι περιοχές που θα δώσει δεν θα είναι επακριβώς αντικείμενα **αλλά κάποιες φορές δυο αντικείμενα μαζί, κομμένα αντικείμενα κτλ.**

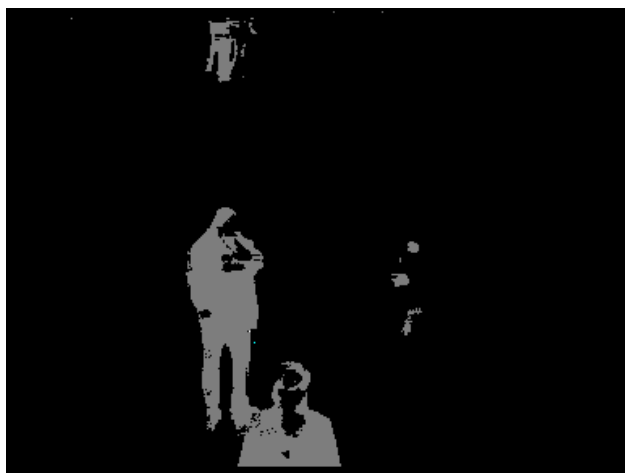
Πχ



**ΕΙΚΟΝΑ 5.2.2.1 ΣΥΓΧΩΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ (video 1)**

Εδώ, έχω ξεχωρίσει το αντικείμενο (ο κάτω άνθρωπος) στην ακολουθία αλλά στην εικόνα βγαίνει και ο άλλος άνθρωπος (λόγω κοινής περιοχής που έχουν). Αντί λοιπόν εδώ να επιλέγω **ακριβώς το κέντρο βάρους (που μπορώ εύκολα να το κάνω στο περίπου)**, αφήνω το σύστημα να επιλέξει κέντρο βάρους της labeled region, με το τύπο που παρουσιάσαμε λίγο παραπάνω. Δημιουργούνται ασυνέχειες και μετατοπίσεις σε αυτό, πολλές φορές αρκετά έντονες

Πχ



**ΕΙΚΟΝΑ 5.2.2 ΜΗ ΙΔΑΝΙΚΟ LABELING (video 2)**

Ειδικά στο 2<sup>ο</sup> video, το περιοδικό freezing θα κάνει ακόμα πιο δύσκολη την πρόβλεψη διότι αν στο frame πριν το freezing δίνει μια πρόβλεψη κ ταχύτητας, η ταχύτητα στο frame με freezing είναι 0, συνεπώς η πρόβλεψη λάθος, και το φίλτρο «επαναπροσδιορίζει» τα βάρη του ώστε να συγκλίνει παρακάτω κτλ

Έτσι, από αυτά αναμένουμε χειρότερη απόδοση στο video 2, που όμως αν γενικά κινηθεί σε λογικά πλαίσια ( μια μέση απόκλιση 5-10 pixels της πρόβλεψης από τη πραγματική θέση είναι ανεκτή, ο τύπος του tracking εδώ δεν απαιτεί μεγάλη ακρίβεια), θα έχουμε επιτύχει κάτι σημαντικό, να επιλύσουμε το πρόβλημα του tracking κάτω από τις δύσκολες αυτές συνθήκες – φωτισμού, άσχημης λήψης ανθρώπινη κίνηση άρα μη γραμμική κτλ- ,με ένα τρόπο σημαντικά πιο γρήγορο και απλούστερο από το να καταφύγουμε σε EKF και άλλες, πολύπλοκες υπολογιστικά, τεχνικές.

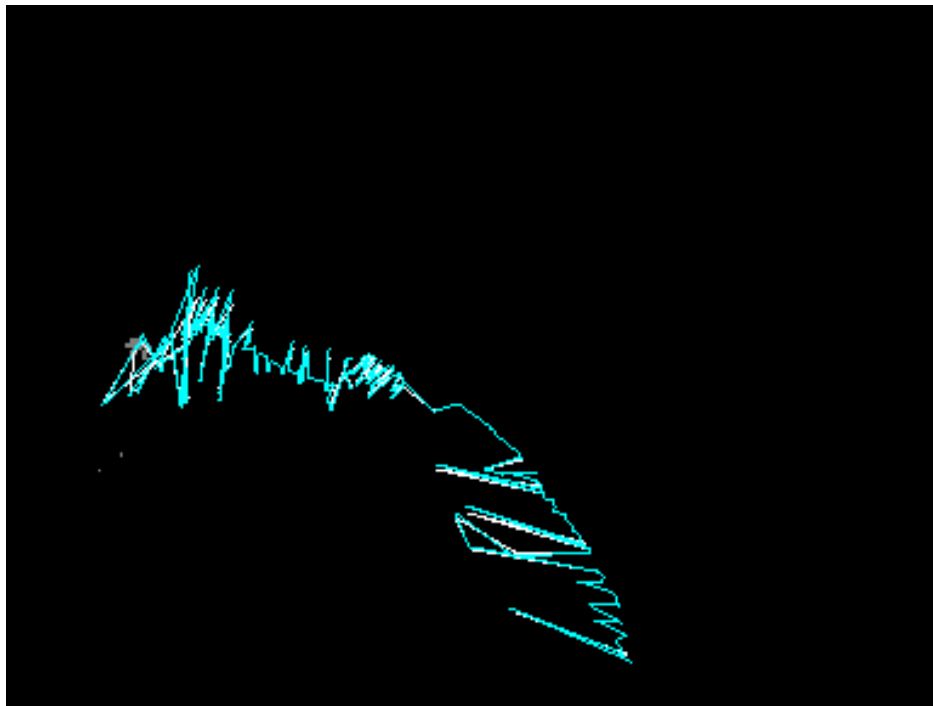
- Παράλληλα, όπως είδαμε, για καλύτερη εποπτεία του αποτελέσματος, απομονώνουμε και κρατάμε μόνο τη περιοχή του αντικειμένου σε κάθε frame, ενώ θέτουμε μια κοινή ένταση σε όλες τις περιοχές αυτές στο video.

-Ξεκινάμε λοιπόν, με τα παραπάνω να κάνουμε kalman filtering και να ελέγχουμε τα αποτελέσματα του στη πρόβλεψη του σημείου κέντρου βάρους που ορίσαμε ως άσπρη κουκίδα. Σε κάθε frame n, σημειώνω τη πρόβλεψη (που έχει από το προηγούμενο frame) με {0,255,255} (Ανοιχτό Πράσινο Του Νεφρίτη) RGB

### 5.3 ΑΠΟΤΕΛΕΣΜΑΤΑ

#### 1)VIDEO 1

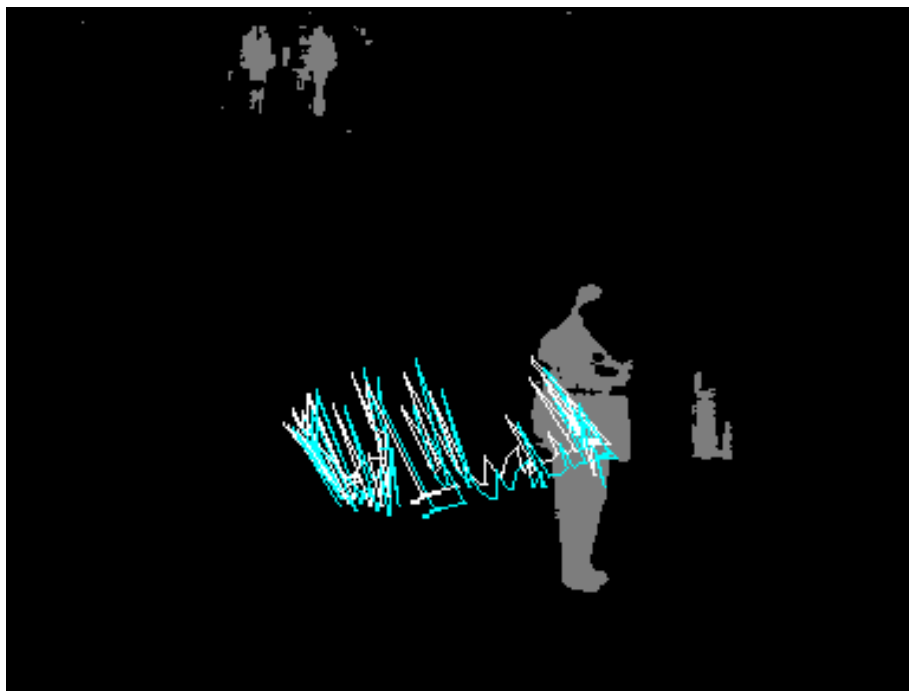
Όπως περιμέναμε, τα αποτελέσματα εδώ είναι αρκετά καλά, η πρόβλεψη αποκλίνει κατά μέσο όρο περίπου 1-2 pixels από την πραγματική θέση



**ΕΙΚΟΝΑ 5.3.1 ΤΟ ΜΟΝΟΠΑΤΙ ΠΟΥ ΑΚΟΛΟΥΘΕΙ Η ΠΡΟΒΛΕΨΗ ΜΕ  $RGB\{0,255,255\}$  ΚΑΙ ΤΟ ΜΟΝΟΠΑΤΙ ΤΗΣ ΠΡΑΓΜΑΤΙΚΗΣ ΚΙΝΗΣΗΣ  $RGB\{255,255,255\}$  ΠΑΡΑΤΗΡΟΥΜΕ ΠΩΣ ΠΡΟΣΕΓΓΙΖΟΥΝ ΑΡΚΕΤΑ. Η ΠΡΩΤΗ ΠΑΡΟΥΣΙΑ DISTURBANCES ΣΤΗ ΚΙΝΗΣΗ ΚΑΤΑΔΕΙΚΝΥΕΙ ΤΗ ΣΤΡΟΦΗ ΠΟΥ ΠΑΙΡΝΕΙ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΩΣΤΕ ΝΑ ΣΥΝΑΝΤΗΣΕΙ ΤΟ ΑΛΛΟ ΑΤΟΜΟ ΠΑΡΑΠΕΡΑ, Η 2<sup>Η</sup> ΤΗ ΣΥΓΧΩΝΕΥΣΗ ΤΟΥ ΜΕ ΤΟ ΑΛΛΟ ΑΤΟΜΟ (ΧΕΙΡΑΨΙΑ).**

Το σύστημα μας εδώ αποδίδει αρκετά καλά και **ας μην είναι στη θεωρία ου προορισμένο για τέτοιες ασυνέχειες στην κίνηση**. Σ' αυτό το βοηθάει η απουσία φαινομένων όπως πχ το freezing που θα δούμε στο 2<sup>ο</sup> video, αλλά και ότι τα αντικείμενα είναι μικρά και λίγα, συνεπώς οι όποιες αποκλίσεις (όπως προς το τέλος της κίνησης του, όπου για ένα διάστημα «συγχωνεύεται» με τον άλλο άνθρωπο), θα βρίσκονται σχετικά κοντά με την πρόβλεψη (σε μια γειτονιά το πολύ 5-6 pixels).

## **2) VIDEO 2**



**ΕΙΚΟΝΑ 5.3.2 ΤΑ ΑΝΑΛΟΓΑ ΜΟΝΟΠΑΤΙΑ ΠΡΟΒΛΕΨΗΣ ΚΑΙ ΠΡΑΓΜΑΤΙΚΗΣ ΚΙΝΗΣΗΣ. ΟΙ ΑΠΟΚΛΙΣΕΙΣ ΕΔΩ ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΕΣ, ΛΟΓΩ ΤΟΥ FREEZING ΚΑΙ ΤΩΝ ΜΕΓΑΛΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΠΟΥ ΕΧΟΥΜΕ, ΤΑ ΟΠΟΙΑ ΛΟΓΩ ΤΟΥ ΜΗ ΤΕΛΕΙΟΥ LABELING ΚΑΠΟΙΕΣ ΣΤΙΓΜΕΣ ΣΥΓΧΕΩΝΤΑΙ. ΤΑ ΑΠΟΤΟΜΑ PEAKS ΚΑΤΑΔΕΙΚΝΥΟΥΝ ΤΑ FREEZING FRAMES, ΕΝΩ ΓΙΝΟΝΤΑΙ ΠΙΟ ΕΝΤΟΝΑ ΑΝ ΕΧΩ ΚΑΙ ΣΥΓΧΩΝΕΥΣΗ ΑΛΛΗΣ REGION ΜΕ ΑΥΤΗ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΤΟ FRAME**

Εδώ, τα αποτελέσματα είναι αρκετά χειρότερα **αλλά κινούνται σε ιδιαιτέρως ικανοποιητικά πλαίσια** (για τη δυσκολία του video, τις απαιτήσεις μας κτλ) με μέσο όρο απόκλισης πρόβλεψης/πραγματικότητας τα 5-6 pixels.

Συνολικά, θα πρέπει να πούμε ότι στις περιοχές που έχω γενικά γραμμική κίνηση και όχι απότομες μεταβολές το kalman συμπεριφέρεται **ιδιαιτέρως καλά και στα δυο video, με την όποια απόκλιση να οφείλεται περισσότερο στη συμμετοχή των προηγούμενων απότομων μεταβολών στη διασπορά, μέση τιμή κτλ του μοντέλου κίνησης.**

#### 5.4 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΣΥΝΟΛΙΚΗ ΚΡΙΤΙΚΗ TRACKING

Υλοποιήσαμε μια από τις πολλές μορφές kalman filtering, την κατά γενική ομολογία ευκολότερη και λιγότερο δαπανηρή υπολογιστικά, συνολικά κάθε prediction-update iteration δεν ξεπέρασε το 1-2 sec. Θεωρήσαμε ασυσχέτιστες τις δυο διαστάσεις κίνησης, γραμμική κίνηση, την προσέγγιση του  $u_n = 3 y_n \text{Var}(y_n)$  ως by-the-book υπόδειξη από την εμπειρία ειδικών [36.ii]. Αυτό το οποίο ξεχωρίσαμε εμείς είναι αυτή η μη-γραμμική «μετατροπή» του workload και το τόλμημα μας να το εξετάσουμε με τη χρήση αυτού του σχετικά απλά ορισμένου kalman filter.

Τα αποτελέσματα μας δικαιώνουν όμως, αν και δεδομένα οι μη- γραμμική κίνηση δημιουργεί αποκλίσεις από το επιθυμητό αποτέλεσμα, αυτό κινείται σε πολύ ικανοποιητικά πλαίσια λαμβάνοντας υπόψη και ότι στο συγκεκριμένο σύστημα οι απαιτήσεις ακρίβειας δεν είναι πολύ αυστηρές.

Το kalman filter λοιπόν αποδεικνύει την ισχύ του ως predictor (χρησιμοποιείτε σε πολλά πεδία, πέρα από computer vision στην ανάλυση κυκλωμάτων κτλ). Παρά την ηλικία του (ο Kalman διατύπωσε την αρχική του μορφή κάπου στα 1960) αποτελεί ένα πεδίο πολύ ενεργό στην έρευνα, ένα εργαλείο – ειδικά πιστεύουμε αν στο μέλλον ασχοληθούμε με το μη γραμμικό EKF και τις παραλλαγές του- που πιστεύουμε πως μπορεί να παίξει σπουδαίο ρόλο σε όλα τα πεδία που απαιτούν πρόβλεψη, είτε αυτόνομα είτε σε συνδυασμό με τις πιο ευρέως διαδεδομένες τεχνικές όπως AI agents, FUZZY LOGIC, Neural Networks κτλ. Στην εργασία μας δείχνει την ισχύ του στην πιο απλή και προσεγγιστική μορφή και αναμένουμε σε ανώτερο επίπεδο (EKF, συνδυασμός με NN κτλ) να καταδειχθεί η ισχύ του περαιτέρω στο μέλλον. Ήδη αποτελεί το πλέον διαδεδομένο εργαλείο πρόβλεψης στην τεχνολογία surveillance (είτε για στρατιωτικούς σκοπούς είτε πολιτικούς) και αυτό πιστεύουμε είναι πολύ σημαντικό.

## **ΚΕΦΑΛΑΙΟ 6: ΕΠΙΛΟΓΟΣ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ**

- Γενικές Ιδέες Που Προέκυψαν
- Σημειώσεις Για Περαιτέρω Έρευνα

## 6. ΕΠΙΛΟΓΟΣ

Οι Forsyth – Ponce στο βιβλίο τους Computer Vision: A Modern Approach [39] χαρακτηρίζουν γενικά το πεδίο της τεχνητής όρασης ως ένα «Σύννορο», μια επιστήμη που αλληλεπιδρά με πλήθος άλλες και που οι τεχνικές της και η μεθοδολογία της βασίζονται ακόμα κατά πολύ στην εμπειρία και λιγότερο στην επιτηδευμένη μαθηματική διατύπωση. Και πραγματικά, πολλές φορές το παρόν πόνημα καθυστέρησε επειδή αντιμετωπίσαμε δυσκολίες στην κατανόηση πολύπλοκων μαθηματικών εννοιών. Προχωρήσαμε τότε και μόνο τότε, όταν αφήσαμε κάπως κατά μέρους τα πολλά μαθηματικά και επικεντρωθήκαμε στη «φυσική» σημασία των διεργασιών, οδηγούμενη τελικά στο σωστό αποτέλεσμα, με όχι απαραίτητα πάντα το πιο δύσκολο ή politically correct τρόπο.

Έτσι λοιπόν, θεωρούμε πως η εργασία αυτή συμβάλει στο ότι κατ' αρχάς συγκεντρώνει το σύνολο έρευνας από πολλές και διαφορετικές πηγές πάνω στο αντικείμενο του change detection και του object tracking, με αρκετά πιστεύουμε απλό και κατανοητό τρόπο, δίνοντας περισσότερο σημασία στη πράξη παρά στην αυστηρή θεωρία. Σε συνδυασμό με την δική μας υλοποίηση προκύπτει μια μεθοδολογία αντιμετώπισης ( προεπεξεργασία και illumination invariant filtering με homomorphic  $\rightarrow$  Euler Number  $\rightarrow$  Labeling  $\rightarrow$  Kalman filtering) , που δύναται να δώσει ποιοτικά αποτελέσματα αυτούσια ή τμήματα της να χρησιμοποιηθούν προς βελτίωση τεχνικών άλλων ερευνητών στο αντικείμενο. Παράλληλα, παρουσιάσαμε μια αρκετά αντιπροσωπευτική μοντελοποίηση του ελέγχου των αποτελεσμάτων σε change detection ( κριτική και επισημάνση των σημαντικών ιδιοτήτων των videos, ορισμός μετρικών σύγκρισης, υλοποίηση της σύγκρισης και «αποκρυπτογράφηση» των αποτελεσμάτων, κεφ.4) που μπορεί να ακολουθηθεί ώστε να δώσει αρκετά αξιόπιστα αποτελέσματα. Ακόμα, «προτείνουμε» το φίλτρο kalman, δείχνοντας την ισχύ του στην πλέον ίσως απλούστερη μαθηματικά μορφή του, ως έναν ισχυρό predictor μηχανισμό όχι μόνο για προβλήματα surveillance και computer vision εν γένει, **αλλά ως ένας μηχανισμός πρόβλεψης όπου τέτοιος είναι απαραίτητος ( από προβλέψεις χαρακτηριστικών κυκλωμάτων, τάση κτλ, έως Agent Searching στο διαδύκτιο και congestion controlling σε δίκτυα), αυτόνομα ή σε συνδυασμό με τις πιο διαδεδομένες τεχνικές, όπως AI, NN κα.**

Ωστόσο βεβαία, δεν ήταν δυνατόν να καλύψουμε εδώ το πολύ μεγάλο όγκο όλων των τεχνικών change detection/ object tracking ( στο 2<sup>ο</sup> μάλιστα, δώσαμε κατ' ευθείαν τη προσοχή μας στο kalman filter, και το κύριο βάρος των επιλογών μας έπεσε στις διαφοροποιήσεις στους αλγόριθμους του change detection). Τεχνικές όπως το EKF για nonlinear motion tracking ή adaptive change detection ( με συνδυασμό αποτελεσμάτων δυο ή περισσότερων τεχνικών), δυναμικό background maintenance (wallflower) κα, καλύφθηκαν από επιφανειακά έως καθόλου, αφού είτε λόγω του όγκου της συνολικής δουλειάς (περισσότερο σε επίπεδο δοκιμών και υλοποίησης), είτε γιατί θεωρήσαμε πως ξεφεύγουν από το σκοπό της παρούσας εργασίας, αναφέρθηκαν αρκετά επιγραμματικά. Ελπίζουμε να ασχοληθούμε με αυτά στο μέλλον, μαζί με άλλα ζητήματα, όπως κυρίως τη μετατροπή του συστήματος μας σε real-time. **Γι' αυτό το σκοπό, κύριο πρόβλημα αποτελεί η χρονική διάρκεια του Euler Change Detection, που θα μπορούσε να λυθεί πχ με χρήση Artificial Intelligence μεθοδολογίας για περιορισμό του εξαντλητικού searching που κάνει σε όλα τα thresholds.** Ακόμα, η αντικατάσταση του kalman filter με EKF, η «αυτοματοποίηση» της εύρεσης της πραγματικής θέσης  $y_n$  στο tracking, ο διαχωρισμός πεπλεγμένων αντικειμένων στο labeling κα είναι ζητήματα προς μελλοντική έρευνα.

## **APPENDICES**

- Error Complementary Function**
- Error Estimation και LMEDS**
- Homomorphic Filtering**
- Μετατροπή Grayscale**

## APPENDICES

### A. ERFC FUNCTION

#### ΜΑΘΗΜΑΤΙΚΗ ΕΙΣΑΓΩΓΗ

##### 1. ΟΡΙΣΜΟΣ

Έστω μια τιμή  $x \in \mathbb{R}$ . Η  $\text{erfc}(x)$  (Error Complementary Function) ορίζεται μαθηματικά ως η λύση της διαφορικής εξίσωσης

$$2x \frac{dy(x)}{dx} + \frac{d^2 y(x)}{dx^2} = 0 \quad (\text{A.1})$$

με αρχικές συνθήκες επίλυσης

$$\text{erfc}(0)=1, \quad \frac{d\text{erfc}(x)}{dx}(0) = \frac{-2}{\sqrt{\pi}} \quad (\text{A.2})$$

##### 2. ΑΝΑΛΥΣΗ ΣΕ ΣΕΙΡΑ TAYLOR ΚΑΙ ΕΠΙΛΥΣΗ

Η ανάπτυξη σε σειρά Taylor δίνει τη μορφή

$$\begin{aligned} \text{erfc}(x) = & \frac{-2}{\sqrt{\pi}}x + \frac{-2}{3\sqrt{\pi}}x^3 - \frac{1}{5\sqrt{\pi}}x^5 + \frac{1}{21\sqrt{\pi}}x^7 - \frac{-1}{108\sqrt{\pi}}x^9 + \frac{1}{660\sqrt{\pi}}x^{11} \\ & - \frac{1}{4680\sqrt{\pi}}x^{13} + \frac{1}{37800\sqrt{\pi}}x^{15} + O(x^{16}) + 1 \quad (\text{A.3}) \end{aligned}$$

δηλ.  $\text{erfc}(x) = \sum_{n=0}^{\infty} u(n)x^n$ , με  $2nu(n) + (n^2 + 3n + 2)u(n+2) = 0$  και

$$u(1) = \frac{-2}{\sqrt{\pi}}, \quad u(0) = 1 \quad (\text{A.4})$$

Σε μορφή με φθίνων πολυώνυμο ε το A.3 δύναται να εκφραστεί ως

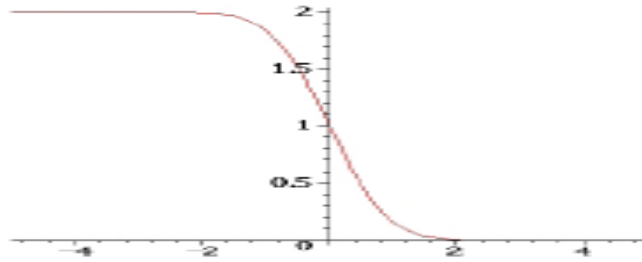
$$\text{erfc}(x) = e^{\frac{-1}{x^2}}(xy_0(x)), \quad \text{με } y_0(x) = \frac{1}{\sqrt{\pi}} - \frac{x^2}{2\sqrt{\pi}} + 2... \quad (\text{A.5})$$

με αρχικές συνθήκες  $u(1) = 0, u(0) = \frac{1}{\sqrt{\pi}}$

και  $2nu(n) + u(n-2)(-4 + 3n + (n-2)^2) = 0$ .

Από το όνομα της ακόμα μπορούμε να καταλάβουμε πως έχει μορφή μιας πιθανοτικής κατανομής, πράγματι το γράφημα της





**ΕΙΚΟΝΑ Α.1 ERFc(X)**

Για  $x$  θετικά έχει τιμές από 0 έως 1 στο διάστημα  $x \in [0 \dots 2]$ .

### 3.ΥΛΟΠΟΙΗΣΗ ΣΤΗΝ ΕΡΓΑΣΙΑ

Ως συνάρτηση απείρων όρων το πρόβλημα πάντα είναι η επιλογή του αριθμού των όρων της συνάρτησης που πρέπει να κρατήσω ώστε να έχω ένα αποδεκτά μικρό σφάλμα. Έστω η μορφή A.5, την οποία και υλοποίησα. Η επιλογή μου ήταν να κρατήσω τους 15 πρώτους όρους, περισσότερο διαισθητικά και το τελικό αποτέλεσμα στην υλοποίηση του Gaussian change detection μου δείξε ότι ήταν καλή επιλογή.

### B.NOISE ESTIMATION-LMEDS

Η προσέγγιση της τιμής της διασποράς του θορύβου σε μια εικόνα είναι πάντα σημαντική και χρήσιμη διαδικασία. Μια προφανή λύση, η χρήση κάποιου διαδεδομένου φίλτρου αποθορυβοποίησης (πχ averaging ή median) και ο υπολογισμός της διασποράς στις τιμές έντασης **της εικόνας που προκύπτει από τη διαφορά μεταξύ Αρχικής εικόνας και denoised εικόνας**. Ακόμα, μπορώ να κάνω την υπόθεση πως οι πολύ μικρές τιμές έντασης της εικόνας είναι μόνο θόρυβος και άρα αφού υπολογίσω ιστόγραμμα εντάσεως της εικόνας να πάρω διασπορά για πχ τις πιο μικρές εντάσεις που έχω που αποτελούν ένα ποσοστό (πχ 5%) αυτής. Η πρώτη μέθοδος, αν και απλή, δίνει σχετικά καλά αποτελέσματα, ενώ υπάρχουν και άλλες, πιο sophisticated προσεγγίσεις που αποδίδουν καλύτερα ή χειρότερα [6].

Στη βιβλιογραφία του change detection [4] αρκετά δημοφιλής μέθοδος παρουσιάζεται η LMedS ,Least Median Of Squares, η οποία είναι αρκετά απλή και έχει καλό αποτέλεσμα, αρκεί ο θόρυβος στην εικόνα να μην ξεπερνά το 50% σ' αυτήν, πράγμα που γενικά μπορούμε να θεωρήσουμε πως ισχύει στη δική μας περίπτωση (ιδιαίτερα για τα image differences, στα οποία και την εφαρμόζουμε).

Η όλη διαδικασία συνίσταται κατά πρώτον στην εύρεση του ιστογράμματος της εικόνας και στη συνέχεια στο sorting αυτού κατά αύξοντα αριθμό pixels που παρουσιάζονται σε κάθε ένταση. Στη συνέχεια, ο αλγόριθμος είναι ο εξής

Κράτησε τη τιμή  $\chi^*$  , υπολογιζόμενη από το sorted ιστόγραμμα  $\{x_{i=a \dots b}\}$  για την οποία έχω τον minimum της σχέσης

$$\text{median} (x^* - x_i)^2 \quad (\text{B.1})$$

$$\text{με } \chi^* = (x_{t+b/2} + x_t) \frac{1}{2}, \text{ όπου } t = \arg \min (x_{t+b/2} + x_t) \quad (\text{B.2})$$

$$\text{τότε, } \sigma = \chi^* / 0.33724 \quad (\text{B.3}) \quad [5]$$

Το μόνο πρόβλημα που προκύπτει είναι το εξής, η εφαρμογή που έχουμε εμείς είναι πάνω σε image differences, συνεπώς το ιστόγραμμα που προκύπτει θα έχει πολλές τιμές έντασης 0, άρα δεν αποκλείω τελικά αν πάρω  $a=0$  και  $b=255$  να έχω  $\sigma=0$ . Αυτό αν το περάσω στο τύπο υπολογισμού της Gaussian παραπάνω να δώσει  $\text{erfc}(\infty)$ , που είναι λάθος. Η λύση ακριβώς βασίζεται στην απλή λογική, ο θόρυβος που απομένει στο image differences ΔΕΝ ΕΙΝΑΙ ΠΡΟΣΘΕΤΙΚΟΣ σε όλες τις εντάσεις. Άρα πρέπει  $a>0$  και  $b=255$ . Εμείς επιλέξαμε  $a$  από εκεί που στο sorted ιστόγραμμα έχω μη μηδενικές τιμές, δηλ. με αυτό τον τρόπο έχω την πιο «απαισιόδοξη» πρόβλεψη για το θόρυβο (max τιμή αυτού) μέσω LMedS.

## C.HOMOMORPHIC FILTERING

Η θεωρητική παρουσίαση του τι είναι homomorphic filtering έγινε παραπάνω στη θεωρία για το change detection, ενώ προσθέσαμε / επαναλάβαμε κάποια πράγματα στη παρουσίαση του αλγόριθμου του Aach. Εδώ απομένει να αναλύσουμε κάποιες τεχνικές υλοποίησης και να δείξουμε ποια και γιατί επιλέξαμε εμείς.

Γενικά, υπάρχουν τρεις στρατηγικές που μπορούμε να ακολουθήσουμε, περισσότερα για τα παρακάτω μπορούν να βρεθούν στο [38]

### 1) SPECTRAL DOMAIN ΥΛΟΠΟΙΗΣΗ

Όπως είδαμε στη θεωρία, η τιμή έντασης σε ένα pixel μπορεί να εκφραστεί ως το γινόμενο της φωτεινότητας (illumination) και της αντανάκλασης (reflectance)

$$f(x,y,z)=i(x,y,z)r(x,y,z) \quad (C.1)$$

$$\text{θέτοντας } g(x,y,z)=\ln(f(x,y,z))=\ln(i(x,y,z))+\ln(r(x,y,z)) \quad (C.2)$$

$$\text{τότε } F\{g(x,y,z)\}=F\{\ln(f(x,y,z))\}+F\{\ln(r(x,y,z))\} \quad (C.3)$$

$$\text{ή } G(u,v,w)=I(u,v,w)+R(u,v,w) \quad (C.4)$$

με  $F$  να συμβολίζει το μετασχηματισμό Fourier.

Τώρα, με την εφαρμογή ενός φίλτρου  $H(u,v,w)$  (εξηγούμε παρακάτω)

$$S(u,v,w)=H(u,v,w)G(u,v,w)=H(u,v,w)I(u,v,w)+H(u,v,w)R(u,v,w) \quad (C.5)$$

Και παίρνοντας τον αντίστροφο μετασχηματισμό Fourier  $F^{-1}$

$$s(u,v,w)=F^{-1}\{S(u,v,w)\}=F^{-1}\{H(u,v,w)I(u,v,w)\}+F^{-1}\{H(u,v,w)R(u,v,w)\} \quad (C.6)$$

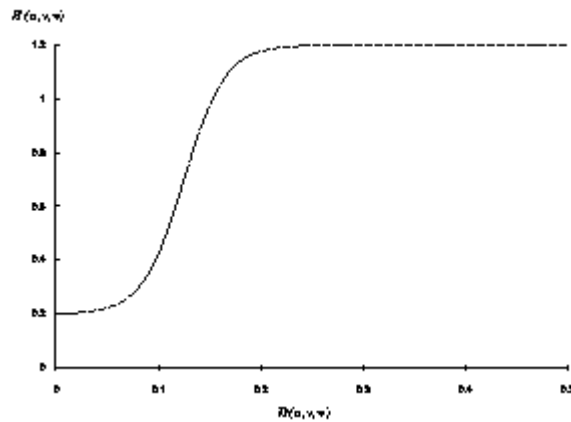
και για να πάρουμε μόνο το reflectance component υψώνω  $\exp[s(u,v,w)]$

$$\exp[s(u,v,w)]=s'(u,v,w)=\exp[i'(x,y,z)]+\exp[r'(x,y,z)] \quad (C.7)$$

$$\text{δηλ. } s'(u,v,w)=i''(x,y,z)+r''(x,y,z)$$

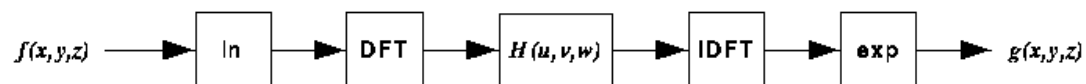
Όπου  $r''$  και  $i''$  οι reflectance και illumination components της τελικής εικόνας αντίστοιχα. Είπαμε στη θεωρία πως το illumination component τείνει να είναι περίπου σταθερό σε όλη την εικόνα, ενώ το reflectance (που είναι κατ' ουσία η

πραγματική πληροφορία στην εικόνα) διαφέρει σημαντικά από περιοχή σε περιοχή στην εικόνα. Άρα, αν το  $H$  ένα high pass frequency filter τότε μπορώ τελικά να έχω  $i''(x,y,z) \rightarrow 0$ . Παράδειγμα για το  $H$



### EIKONA C.1 HIGH PASS FREQUENCY FILTER

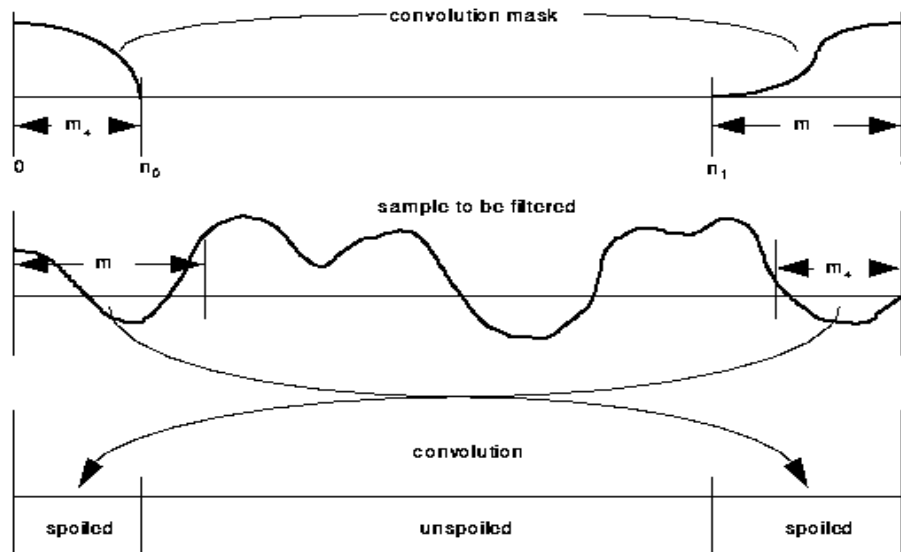
Και η συνολική μέθοδος σε block diagram



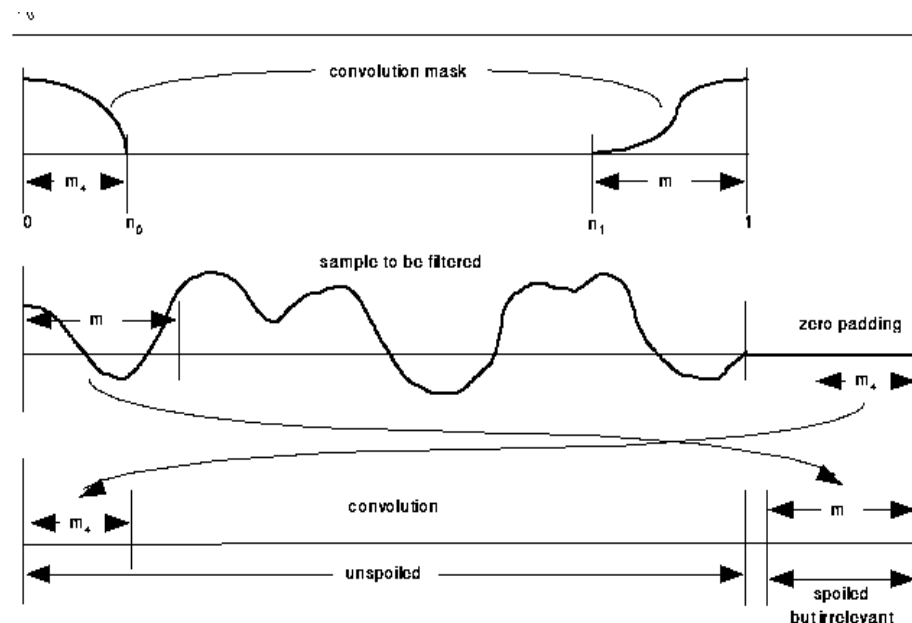
### EIKONA C.2 SPECTRAL DOMAIN HOMOMORPHIC FILTERING

Η μέθοδος αυτή παρουσιάζει κάποια σημαντικά μειονεκτήματα. Αφενός μεν, έχει υψηλό υπολογιστικό κόστος, ειδικά αν αναλογιστούμε εφαρμογές σε μεγάλων διαστάσεων εικόνες ή multidimensional intensities. Αφετέρου δε, η μέθοδος παρουσιάζει και χαμηλή απόδοση. Αυτό συμβαίνει λόγω ενός φαινομένου που παρουσιάζεται στο spectral domain, το circular convolution. Ο πολλαπλασιασμός (C.5) στο επίπεδο συχνοτήτων μετατρέπεται σε convolution στο πραγματικό επίπεδο. Το αποτέλεσμα του (C.5) είναι μια εικόνα μεγαλύτερων διαστάσεων από την αρχική  $f$ , η επιπλέον αυτή πληροφορία εμφανίζεται στα άκρα της εικόνας αποτελέσματος (και άρα αλλοιώνει το πραγματικό αποτέλεσμα).

Αυτό το φαινόμενο, είναι μια ιδιότητα που εμφανίζεται γενικά στο filtering σε επίπεδο συχνοτήτων. Μπορεί να αποφευχθεί με χρήση zero-padding στα άκρα της αρχικής εικόνας και αφαίρεση του τμήματος αυτού στο τέλος της διαδικασίας. Αλλά και αυτό μεγαλώνει περαιτέρω το υπολογιστικό κόστος.



ΕΙΚΟΝΑ C.3.1 ΕΜΦΑΝΙΣΗ ΤΟΥ ΦΑΙΝΟΜΕΝΟΥ DISTORTION



ΕΙΚΟΝΑ C.3.2 ΕΠΙΛΥΣΗ ΤΟΥ ΦΑΙΝΟΜΕΝΟΥ ΜΕΣΩ ZERO PADDING

## 2) SPATIAL DOMAIN ΥΛΟΠΟΙΗΣΗ

Μπορούμε να βελτιώσουμε τα προβλήματα distortions και υπολογιστικού κόστους της παραπάνω μεθόδου, με το να έχουμε υπολογίσει από πριν τον αντίστροφο μετασχηματισμό Fourier του  $H(u,v,w)$

$$h(x, y, z) = F^{-1}[H(u, v, w)] \quad (C.8)$$

τότε, μπορούμε να εφαρμόσουμε το φίλτρο μέσω convolution, και από τις C.5, C.6 και C.7 έχω

$$s(x, y, z) = \exp[h(x, y, z) * \ln(f(x, y, z))] \quad (C.9)$$

Η μέθοδος αυτή βελτιώνει την απόδοση αλλά επιπλέον έχει το πρόβλημα ότι έχει μεγάλο υπολογιστικό κόστος, αν η  $h(x,y,z)$  ( convolution mask) είναι μεγάλη. Φυσικά, υπάρχουν μέθοδοι για προσέγγιση της με convolution mask μικρών διαστάσεων.

### 3) ΥΛΟΠΟΙΗΣΗ ΜΕ LOW PASS FILTER ΣΤΟ ΠΡΑΓΜΑΤΙΚΟ ΕΠΙΠΕΔΟ

Η μέθοδος αυτή είναι απλή, με σχετικά χαμηλό υπολογιστικό κόστος και αποτελεί την δική μας υλοποίηση του homomorphic filtering.

Επανεκφράζοντας τη σχέση illumination- reflectance components με το συνολικό intensity στην εικόνα έχω

$$f(x, y, z) = i_0 \cdot i(x, y, z) \cdot r(x, y, z) \quad (C.10)$$

όπου  $i_0$ , το επιθυμητό ποσοστό του illumination που θέλουμε να κρατήσουμε (ώστε να μην έχω εντελώς σκοτεινό αποτέλεσμα,  $i$  τα illumination variations που θέλω να αφαιρέσω από την εικόνα και  $r$  το reflectance που έχει και την πραγματική πληροφορία της εικόνας.

Από αυτά μπορώ να έχω

$$g(x, y, z) = \ln(f(x, y, z)) = \ln i_0 + \ln(i(x, y, z)) + \ln(r(x, y, z)) \quad (C.11)$$

αφού το  $i$  μεταβάλλεται ελάχιστα στην εικόνα ενώ το  $r$  μεταβάλλεται περισσότερο, αν κάνω low pass filtering στο  $g$  μπορώ να απομονώσω το  $i$ .

$$g'(x, y, z) = LPF\{g(x, y, z)\} = \ln(f(x, y, z)) = \ln i_0 + \ln(i(x, y, z)) \quad (C.11)$$

$$\text{και } s(x, y, z) = g(x, y, z) - g'(x, y, z) + \ln i_0 \simeq \ln i_0 + \ln(r(x, y, z)) \quad (C.12)$$

τελικά, υψώνοντας τη C.12 σε δύναμη του  $e$  έχω το ζητούμενο  $r$

$$s'(x, y, z) = \exp[s(x, y, z)] \simeq i_0 r(x, y, z) \quad (C.13)$$

Το low pass filtering μπορεί να υλοποιηθεί με neighborhood averaging σε κάθε θέση pixel

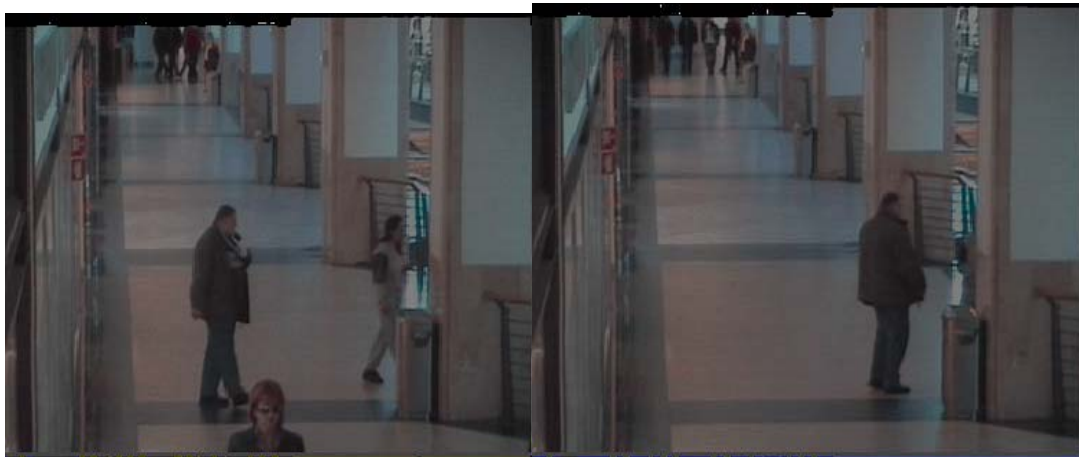
$$LPF[g(x,y,z)] \simeq \frac{1}{N_x + N_y + N_z} \sum_{i=x-N_x}^{x+N_x} \sum_{j=y-N_y}^{y+N_y} \sum_{k=z-N_z}^{z+N_z} g(i, j, k) \quad (C.14)$$

Στην δική μας υλοποίηση  $N_z=3$ , δηλ. χρησιμοποιήσαμε ένα 3X3 averaging filter  
Εδώ χρειάζεται zero padding αλλά γενικά η όλη υλοποίηση είναι αρκετά γρήγορη (συνολικά για κάθε image μέτρησα χρόνο επεξεργασίας πολύ λιγότερο από 1 sec)

Μερικά από τα αποτελέσματα που είχα



**ΕΙΚΟΝΑ C.1 HOMOMORPHIC FILTERING ΣΤΟ 1<sup>ο</sup> VIDEO**



**ΕΙΚΟΝΑ C.2 ΑΠΟΤΕΛΕΣΜΑΤΑ HOMOMORPHIC FILTERING ΣΤΟ 2<sup>ο</sup> VIDEO**

Γενικά καλά αποτελέσματα , ένα μικρό μειονέκτημα μόνο κάποια block artifacts στο 1<sup>ο</sup> video στην περιοχή του πολύ έντονου εξωτερικού φωτισμού (τζαμαρία) – λόγω των υπερβολικών μεταβολών του φωτός εκεί- που μπορεί να διορθωθεί πχ με χρήση median filtering

#### **D. GRAYSCALING ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ**

Αν και υπάρχουν αρκετές προσεγγίσεις για τον συνδυασμό της RGB πληροφορίας σε ένα μόνο grayscale κανάλι ένας γενικά αποδεκτός συνδυασμός –ως προς τη ποιότητα του αποτελέσματος του- είναι

$$\text{Grayscale} = (222R + 707G + 71B)/1000$$

Τον οποίο και χρησιμοποιήσαμε



**EIKONA D.1 GRAYSCALING ΣΤΑ VIDEOS**

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

**0. Image Change Detection Algorithms : A systematic Survey, R.J.Radke – S.Adra – O.Al Kofani – B.Roysam, Renssler Polytechnic – Troy –USA, 2004**

(Χρησιμοποιήθηκε ως γενικό guideline και για την εύρεση αρκετών από τις παρακάτω πηγές )

**1.Classification of change detection algorithms for object-based applications, A.Cavallaro-T.Ebrahimi, Signal Processing Laboratory, Lausanne-Switzerland**

**2.Requirements for front-end scene change detection algorithms, E.Stringa-J.Conclaves, IAEA**

**3.Image difference threshold strategies and shadows detection, P.Rosin-T.Elis, Cardiff University, UK**

**4. Robust pixel unmixing, P.Rosin, IEEE Transactions of Geoscience and Remote Sensing, vol.39- No9, September 2001**

**5. Thresholding for Change Detection, P.Rosin, Cardiff University, UK, 2001 (Computer Vision and Image Understanding, vol.86, no.2, 79-95, May 2002)**

**6. Noise variance estimation in images, S.I.Olsen , University Of Copenhagen, Denmark**

**7. Survey over image thresholding techniques and quantitative performance evaluation, M.Sezgin-B.Sankur, Journal of Electronic Imaging, 13(1) , 146-165, January 2004**

**8. Wallflower: principles and practice of background maintenance, K.Toyama-J.Krumm-B.Brummit, Microsoft Research**

9. Change Detection Based on color edges, A.Callavaro-T.Ebrahimi, Signal Processing Laboratory, Lausane-Switzerland
10. Change Detection for Perimeter Surveillance, M.Forshaw-S.Young-M.Hodgetts, Sira Technology Center, Kent-UK
11. Detecting Perceptual Color Changes for Scene Surveillance, M.Rautiainen-T.Ojala-H.Kanniskangas, University of Oulu, **FINLAND**
12. Σημειώσεις Του Μαθήματος Computer Vision, Ε.Πετράκης , Πολυτεχνείο Κρήτης – Χανιά
13. Detection and Recognition of Moving Objects using statistical motion detection and Fourier descriptors, D.Toth-T.Aach, University of Lübeck, Germany, 2003
14. Bayesian Spatio-Temporal Motion Detection Under Varying Illumination, D.Toth-T.Aach-V.Metzler , University Of Lübeck, Germany, 2004
15. Bayesian Algorithms for adaptive detection in image sequences using Markov Random fields, RWTH-Aachen, Germany, 1994 ( Signal Processing: Image Communication, 7(2), 147,60, 1995 )
16. Nonlinear filtering of multiplied and convoled signals, Proc.IEEE, 56(8), 1264-1291, 1968
17. Methods for Correcting lighting pattern and attenuation in underwater imagery, A.Can – H.Singh, IEEE journal of Oceanic Engineering, vol.28 , no.4, 651-672, 2004
18. Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis, S.Negahdaripour, IEEE Trans. Pattern Anal. Machine Intell. , vol.20 , no.9, 961-979, 1998
19. Detecting Changes of Buildings from aerial images using shadow and shading model, in ICPR 98, 1408-1412 ,1998
20. Background and foreground modelling using non-parametric kernel density estimation for visual surveillance, A.Elghammar- R.Duraiswami – D.Harwood – L.S.Davis , Proceedings of the IEEE , vol.90 , no.7, 1151-1163, July 2002
21. Unsupervised Statistical Detection of Changing Objects in Camera-In-Motion video, R.Dahyot – R.Charbonnier – F.Heitz, ICIP-2001 , Salonika – Greece , 2001
22. An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images, IEEE Trans. Image Processing, vol.11, no.4, 452-466, April 2002



23. Maximum likelihood from incomplete data via the EM algorithm, A.P.Dempster-N.M.Laird-D.B.Rubin, Royal Statist. Soc. , vol. 39, no.1, 1-38,1977
24. The reduced Parzen classifier, K.Fukunaga – R.Hayes, IEEE Trans. Pattern Anal. Machine Intell. , vol.11 , April 1989
25. Robustly estimating changes in image appearance, M.J.Black – D.J.Fleet – Y.Yacoob, Computer Vision and Image Understanding , vol.78 , no.1 , 8-31, 2000
26. New likelihood test methods for change detection in image sequences, Y.Z.Hsu – H.Nagel – G.Reckers, Computer Vision Graphics and image processing, vol.26, 73-106 , 1984
27. Adaptive algorithms for change detection in image sequences, A.Elfishawy-S.Kesler – A.Abutaleb, Signal Processing, vol.23, no.2, 179-191, 1989
28. Optimum Multisensor data fusion for image change detection, IEEE Trans. Systems, Man and Cybernetics, vol.25, no.9, 1340-1347, September 1995
29. Detection and analysis of change in remotely sensed imagery with application to wide area surveillance, M.Carlotto, IEEE Trans. Image Processing, vol.6, no.1, 189-202, January 1997
30. Change Detection in overhead imagery using neural networks, C.Clifton, Applied Intelligence, vol.18, 215-234, 2003
31. Illumination independent change detection for real world image sequences, K.Skifstad – R.Jain, Computer Vision Graphics and Image Processing, vol.46, no.3, 387-399, 1989
32. Change Detection by nonlinear Grammian, NSIP 2001, September 2001
33. Bayesian illumination-invariant motion detection, T.Aach – L.Dümbgen – R.Mester – D.Toth, Proc. IEEE , International Conference on Image Processing, October 2001, 640-643
34. Stochastic relaxation, Gibbs Distributions and the Bayesian restoration of images, S. and D. Geman, IEEE Trans. Pattern Anal. Mach. Intelligence, vol.6, no.6, 721-741, 1984
35. Evaluation of Global image thresholding for change detection, P.Rosin – E.Ioannidis, Pattern Recognition Letters, vol.24, no.14, 2345-2356, October 2003
36. Tracking And Kalman Filtering Made Easy, E.Brookner, Wiley – Interscience Publications , 1998
  - i. Κεφ.2, 64-69
  - ii. Κεφ.2, 66
  - iii. Κεφ.1
  - iv. Κεφ.2, παρ. 2.4 – Κεφ.16

- 37. Likelihood-based object detection and object tracking using color histograms and EM, P.Withagen – K.Schutte – F.Groch, university of Amsterdam, the Netherlands, 2002**
- 38. Fully Automatic Segmentation of the brain in MRI, M.Atkins – T.Mackiewicz – K.Whittall, Simon Frazer University, Canada**
- 39. Computer Vision: A Modern Approach, D.Forsyth-J.Ponce, Prentice Hall, 2003**