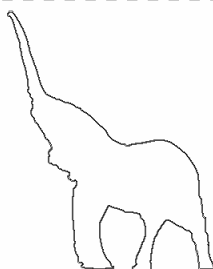




ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
& ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
(Π.Μ.Σ. ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ)

ΑΝΑΚΤΗΣΗ ΠΗΓΩΝ ΒΙΝΤΕΟ ΜΕ ΒΑΣΗ ΤΗ ΣΧΗΜΑΤΙΚΗ ΠΛΗΡΟΦΟΡΙΑ.



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιμέλεια:

Μιχαήλ Α. Κουρουνιώτης

Επιβλέπων:

Δρ. Νικήτας – Μαρίνος Σγούρος
Καθηγητής Πανεπιστημίου Πειραιώς

Πειραιάς, Ιούνιος 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Πειραιάς, Ιούνιος 2008

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη, η ανάλυση, η παρουσίαση και τέλος η αξιοποίηση τεχνικών ανάκτησης εικόνων και ακολουθιών εικόνων (βίντεο) με βάση τη σχηματική πληροφορία.

Στα πρώτα κεφάλαια της εργασίας αναλύονται και παρουσιάζονται τεχνικές εξαγωγής χαρακτηριστικών γνωρισμάτων από εικόνες, και τρόποι αξιοποίησης τους με σκοπό την εξαγωγή της σχηματικής πληροφορίας. Στη συνέχεια παρουσιάζονται οι απαιτήσεις αλλά και οι δυνατότητες των υπάρχοντων συστημάτων ανάκτησης εικόνων και βίντεο και αναλύονται σημαντικές τεχνικές υπολογισμού της ομοιότητας μεταξύ εικόνων και βίντεο.

Στο δεύτερο μέρος της εργασίας παρουσιάζεται ή περιγραφή ενός ολοκληρωμένου συστήματος ανάκτησης πηγών βίντεο που υλοποιήθηκε βάση όσων μελετήθηκαν στα προηγούμενα κεφάλαια. Το σύστημα δέχεται ως είσοδο μία εικόνα η οποία περιέχει κάποιο αντικείμενο. Με χρήση τεχνικών εξαγωγής χαρακτηριστικών γνωρισμάτων εξάγεται το περίγραμμα του αντικειμένου. Στη συνέχεια αυτό το περίγραμμα χρησιμοποιείται σαν ερώτηση (query) με σκοπό να ανακτηθούν τα καρέ ενός βίντεο που περιέχουν κάποιο αντικείμενο όμοιο με αυτό της εικόνας. Το βίντεο δίνεται επίσης ως είσοδος στο σύστημα.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Νικήτα Σγούρο που μου ανέθεσε την εργασία αυτή. Επίσης θα ήθελα να ευχαριστήσω τον Διδάκτορα Ιωάννη Ανδρέου για τις πολύτιμες συμβουλές που μου έδωσε καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ.....	7
1.1. Αντικείμενο της Διπλωματικής Εργασίας	7
1.2. Επισκόπηση των κεφαλαίων της εργασίας	8
2. ΕΞΑΓΩΓΗ ΣΧΗΜΑΤΙΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ	9
2.1. Τεχνικές εξαγωγής των χαρακτηριστικών γνωρισμάτων	10
2.1.1. Ανίχνευση ακμών (<i>Edge Detection</i>)	10
2.1.2. Κατάτμηση Εικόνας (<i>Image Segmentation</i>).....	11
2.1.3. Συνδυασμός των τεχνικών.....	17
2.2. Εξαγωγή χαρακτηριστικών γνωρισμάτων από βίντεο.....	18
2.2.1. Χαρακτηριστικά γνωρίσματα κίνησης.....	19
2.2.2. Χαρακτηριστικά γνωρίσματα ήχου.....	20
3. ΑΝΑΚΤΗΣΗ ΠΟΛΥΜΕΣΩΝ ΒΑΣΕΙ ΠΕΡΙΕΧΟΜΕΝΟΥ	21
3.1. Βάσεις Δεδομένων Εικόνων	22
3.2. Ερωτήματα εικόνων.....	23
3.3. Υπολογισμός της ομοιότητας.....	25
3.3.1. μέτρα ομοιότητας χρώματος.....	25
3.3.2. μέτρα ομοιότητας υφής.....	25
3.3.3. Μέτρα ομοιότητας σχήματος	27
3.3.4. Σχεσιακά μέτρα ομοιότητας.....	30
3.4. Συστήματα ανάκτησης.....	31
3.5. Βάσεις Δεδομένων Βίντεο	31
3.6. Ερωτήματα βίντεο	32
3.7. Υπολογισμός της ομοιότητας σε βίντεο	33

4. ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ.....	35
4.1. Διαδικασία εξαγωγής σχήματος από εικόνα.....	36
4.1.1. <i>Mean Filter</i>	37
4.1.2. <i>K-means clustering</i>	38
4.1.3. <i>Erosion – Dilation</i>	40
4.1.4. <i>Object Isolation</i>	44
4.1.5. <i>Outline Extraction</i>	45
4.2. Χρήση του Java Media Framework.....	47
4.3. Το σύστημα ανάκτησης (CV_TOOL).....	47
4.4. Περιγραφή διεπαφής χρήστη.....	50
5. ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	51
5.1. Απαιτήσεις συστήματος.....	51
5.2. Παρουσίαση διεπαφής χρήστη (GUI).....	52
5.3. Πειραματικά Αποτελέσματα.....	55
5.3.1. <i>Αποτελέσματα εξαγωγής σχήματος από εικόνες</i>	56
5.3.2. <i>Αποτελέσματα εξαγωγής σχήματος από βίντεο</i>	59
5.3.3. <i>Αποτελέσματα ανάκτησης βίντεο</i>	61
6. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΟΠΤΙΚΕΣ.....	64

ΠΑΡΑΡΤΗΜΑ

A. Java Documentation.....	66
-----------------------------------	-----------

ΒΙΒΛΙΟΓΡΑΦΙΑ

1

Εισαγωγή.

Στο παρόν κεφάλαιο αναλύεται το αντικείμενο της παρούσας εργασίας (παράγραφος 1.1). Στην παράγραφο 1.2 γίνεται μία σύντομη επισκόπηση των κεφαλαίων της διπλωματικής εργασίας.

1.1 Αντικείμενο της Διπλωματικής Εργασίας

Τα τελευταία χρόνια, το διαρκώς αυξανόμενο πλήθος των διαθέσιμων οπτικών πηγών πολυμέσων οδήγησε στη ανάγκη δημιουργίας συστημάτων που υποστηρίζουν αυτόματη ανάλυση του περιεχομένου ψηφιακών εικόνων και κινούμενων εικόνων (βίντεο). Η εργασία αυτή έχει σκοπό να περιγράψει και στη συνέχεια να παρουσιάσει τρόπους με τους οποίους είναι δυνατή η εξαγωγή σχηματικής πληροφορίας από ψηφιακή εικόνα καθώς επίσης και τη δυνατότητα ανάκτησης πηγών βίντεο με βάση την πληροφορία αυτή.

1.2 Επισκόπηση των κεφαλαίων της εργασίας

Το 1^ο κεφάλαιο είναι η εισαγωγή, στην οποία περιγράφεται συνοπτικά το αντικείμενο της διπλωματικής εργασίας.

Στο 2^ο κεφάλαιο γίνεται μία αναφορά στα χαρακτηριστικά γνωρίσματα των ψηφιακών εικόνων και στον τρόπο με τον μπορούν αυτά να αξιοποιηθούν ώστε να προκύψει χρήσιμη πληροφορία που αφορά την εικόνα ή και την ακολουθία εικόνων (βίντεο).

Το 3^ο κεφάλαιο αναφέρεται στην ανάκτηση εικόνας και βίντεο με βάση το περιεχόμενο. Αναλύονται οι ανάγκες που οδήγησαν στη δημιουργία τέτοιων συστημάτων ανάκτησης και εξετάζονται τα μέτρα ομοιότητας.

Στο 4^ο κεφάλαιο περιγράφονται οι τεχνικές και οι αλγόριθμοι που χρησιμοποιήθηκαν για την ανάπτυξη ενός συστήματος εξαγωγής σχηματικής πληροφορία από βίντεο και ανάκτησης βίντεο με βάση τέτοιου είδους πληροφορία.

Στο 5^ο κεφάλαιο γίνεται η παρουσίαση του συστήματος που αναπτύχθηκε και παρουσιάζονται κάποια ενδεικτικά πειραματικά αποτελέσματα.

Τέλος στο 6^ο κεφάλαιο αναφέρονται ορισμένα συμπεράσματα που προέκυψαν από τη μελέτη και την υλοποίηση της εφαρμογής και ορισμένες προοπτικές εξέλιξης και βελτίωσης που μπορούν να γίνουν στο μέλλον.

2

Εξαγωγή σχηματικής πληροφορίας.

Η αναζήτηση ομοιοτήτων σε βάσεις δεδομένων εικόνων και βίντεο γίνεται όλο και σημαντικότερη, εξαιτίας του διαρκούς αυξανόμενου όγκου των διαθέσιμων εικόνων. Παράλληλα, τα συστήματα ανάκτησης που βασίζονται στο κείμενο γίνονται αναποτελεσματικά, αφού βασίζονται στο χειρωνακτικό σχολιασμό. Τα συστήματα αναζήτησης που βασίζονται στα μετα-δεδομένα χρησιμοποιούν πρόσθετες πληροφορίες για να περιγράψουν αποτελεσματικά τα αντικείμενα πολυμέσων (π.χ. οντολογίες, σημασιολογική περιγραφή κτλ), τα οποία στις περισσότερες περιπτώσεις δεν είναι διαθέσιμα. Ο μόνος αποτελεσματικός τρόπος φαίνεται να είναι η αναζήτηση βασισμένη στην ομοιότητα του περιεχομένου της εικόνας [9].

Οι γενικότερες τεχνικές που παρέχουν εξαγωγή χαρακτηριστικών γνωρισμάτων από μια εικόνα είναι βασισμένες στην επεξεργασία των χαρακτηριστικών σε χαμηλό επίπεδο, όπως π.χ. το ιστόγραμμα, η ανίχνευση ακμών, η υφή και η τμηματοποίηση της εικόνας.

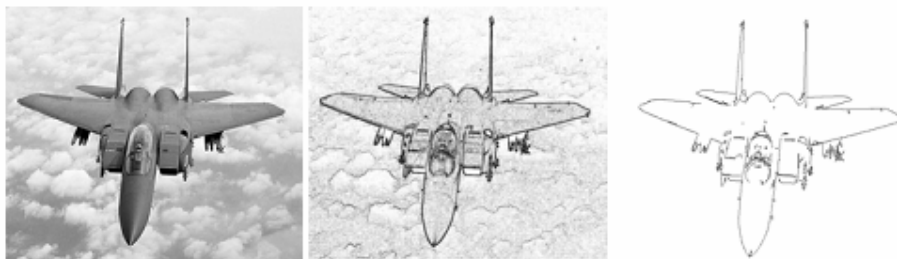
Η εξαγωγή του σχήματος ενός αντικειμένου που βρίσκεται σε μία εικόνα, απαιτεί μία σειρά από βήματα. Ορισμένα από αυτά μπορεί να είναι συνδυασμός των βασικών τεχνικών εξαγωγής των γνωρισμάτων της εικόνας.

2.1 Τεχνικές εξαγωγής των χαρακτηριστικών γνωρισμάτων

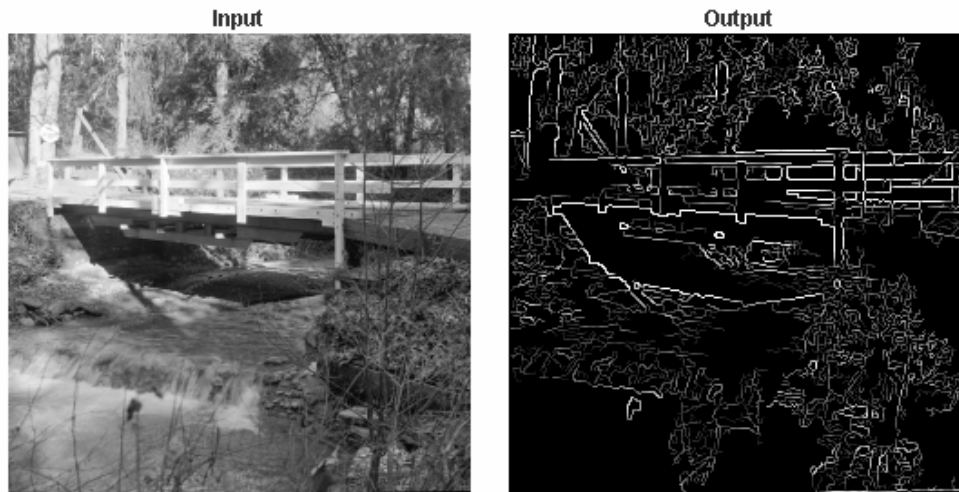
Τα χαρακτηριστικά γνωρίσματα συνήθως αναφέρονται είτε σε απλές δομές όπως τα σημεία ή οι ακμές ή και σε πιο σύνθετες δομές όπως τα αντικείμενα. Άλλα παραδείγματα χαρακτηριστικών γνωρισμάτων συσχετίζονται με την κίνηση σε ακολουθίες εικόνων, με τα σχήματα που καθορίζονται από τις καμπύλες ή τα όρια μεταξύ των διαφορετικών περιοχών εικόνας, ή με τις ιδιότητες μιας περιοχής. Η έννοια χαρακτηριστικά γνωρίσματα είναι πολύ γενική και η επιλογή των χαρακτηριστικών γνωρισμάτων σε ένα συγκεκριμένο σύστημα ψηφιακής όρασης συνήθως είναι εξαρτώμενη από το πρόβλημα. Στη συνέχεια παρουσιάζονται ορισμένες τεχνικές εξαγωγής γνωρισμάτων, που μπορούν να χρησιμοποιηθούν με απώτερο σκοπό την εξαγωγή του σχήματος από μία εικόνα.

2.1.1 Ανίχνευση Ακμών (Edge Detection)

Σε μία ψηφιακή εικόνα, οι ακμές εντοπίζονται στα σημεία που υπάρχουν απότομες εναλλαγές των αποχρώσεων. Οι ακμές χρησιμοποιούνται συχνά για την εξαγωγή σχήματος αντικειμένων από εικόνες. Υπάρχουν αρκετοί αλγόριθμοι για το πρόβλημα της ανίχνευσης ακμών. Μία από τις αποτελεσματικές προσεγγίσεις είναι ο αλγόριθμος Canny [4], που χρησιμοποιεί έναν αλγόριθμο πολλών επιπέδων για να ανιχνεύσει ένα ευρύ φάσμα ακμών σε εικόνες. Τα βήματα του αλγορίθμου είναι η εξομάλυνση της εικόνας με γκαουσιανό φίλτρο, ο υπολογισμός του μεγέθους και του προσανατολισμού κλίσης χρησιμοποιώντας μερικά παράγωγα, εφαρμογή καταστολής nonmaxima στο μέγεθος κλίσης και η χρήση του αλγορίθμου διπλού κατωφλίου για την ανίχνευση και τη σύνδεση των ακμών.



Εικόνα 2.1 – Παράδειγμα εφαρμογής του αλγορίθμου Canny



Εικόνα 2.2 – Παράδειγμα εφαρμογής του αλγόριθμου Canny

Σε αντίθεση με τις περισσότερες τεχνικές που υποθέτουν τον γκαουσιανό θόρυβο και η απόδοσή τους μπορεί να μειωθεί με την παρέκκλιση της κατανομής του θορύβου από το κανονικό, μια διαφορετική προσέγγιση ανίχνευσης ακμών πραγματοποιείται με χρήση εύρωστης στατιστικής (robust statistics) [16]. Η δομή των ακμών ανιχνεύεται αρχικά από ένα εύρωστο μονόδρομο πρότυπο σχεδίου, και εντοπίζεται έπειτα από μια εύρωστη δοκιμή αντίθεσης. Τέλος, εφαρμόζεται κατώφλι υστέρησης για να παραγάγει το χάρτη ακρών.

2.1.2 Κατάτμηση Εικόνας (Image Segmentation)

Η κατάτμηση εικόνας, είναι η διαδικασία κατά την οποία μια ψηφιακή εικόνα χωρίζεται σε ένα αριθμό περιοχών (σύνολα από pixels) που την καλύπτουν. Ο στόχος της κατάτμησης είναι να απλοποιηθεί η εικόνα και να μετατραπεί σε μία μορφή όπου θα είναι ευκολότερη η επεξεργασία της. Η κατάτμηση χρησιμοποιείται για να εντοπίσει αντικείμενα και όρια (γραμμές, καμπύλες, κ.λπ.) σε εικόνες. Οι περιοχές μπορούν επίσης να οριστούν ως οι ομάδες των pixels που έχουν και σύνορα και μια ιδιαίτερη μορφή όπως ένας κύκλος, μια έλλειψη, ή ένα πολύγωνο. Το αποτέλεσμα της κατάτμησης εικόνας είναι ένα σύνολο περιοχών που καλύπτουν ολόκληρη την εικόνα. Κάθε ένα από τα pixels σε μια περιοχή είναι παρόμοιο με τα υπόλοιπα που

ανήκουν στην ίδια περιοχή όσον αφορά κάποια χαρακτηριστικά, όπως το χρώμα, η υφή κλπ.

Η κατάτμηση έχει δύο στόχους. Ο πρώτος στόχος είναι να αποσυντεθεί η εικόνα σε ξεχωριστά μέρη για την περαιτέρω ανάλυση. Σε απλές περιπτώσεις, η διαδικασία κατάτμησης εξάγει μόνο τα μέρη που πρέπει να αναλυθούν περαιτέρω. Παραδείγματος χάριν, η τομή ενός ανθρώπινου προσώπου από ένα καρέ ενός εγχρώμου βίντεο. Η κατάτμηση μπορεί να είναι αξιόπιστη, με την προϋπόθεση ότι το υπόβαθρο δεν έχει τα ίδια χρώματα με ένα ανθρώπινο πρόσωπο. Σε πιο σύνθετες περιπτώσεις, όπως η εξαγωγή ενός πλήρους οδικού δικτύου από μια εναέρια grayscale εικόνα, το πρόβλημα κατάτμησης μπορεί να είναι πολύ δύσκολο και να απαιτήσει την εφαρμογή ακόμα πιο σύνθετων τεχνικών.

Ο δεύτερος στόχος της κατάτμησης είναι να πραγματοποιήσει μία αλλαγή αναπαράστασης. Τα pixels της εικόνας πρέπει να οργανωθούν σε υψηλότερου επιπέδου ομάδες που αξίζουν περαιτέρω ανάλυση. Ένα κρίσιμο ζήτημα είναι εάν η κατάτμηση μπορεί ή όχι να εκτελεστεί για πολλές διαφορετικές περιοχές χρησιμοποιώντας τις γενικές μεθόδους που δεν χρησιμοποιούν οποιαδήποτε ειδική γνώση περιοχών.

Η προοπτική ενός ενιαίου συστήματος κατάτμησης το οποίο θα εργάζεται καλά για όλα τα προβλήματα φαίνεται να είναι αμυδρή. Η εμπειρία έχει δείξει ότι οι εφαρμογές ψηφιακής όρασης θα πρέπει να είναι σε θέση να επιλέξουν από ένα σύνολο εργαλείων και μεθόδων και ίσως να προσαρμόσουν μια λύση χρησιμοποιώντας τη γνώση της εφαρμογής [1].

Προσδιορισμός των περιοχών

Το αποτέλεσμα μίας κατάτμησης εικόνας πρέπει να ικανοποιεί τις παρακάτω επιθυμητές ιδιότητες.

- Οι περιοχές μιας κατάτμησης εικόνας πρέπει να είναι ομοιόμορφες και ομοιογενείς όσον αφορά κάποιο χαρακτηριστικό, όπως το χρώμα, την υφή κλπ.
- Το εσωτερικό των περιοχών πρέπει να είναι απλό και χωρίς πολλές μικρές τρύπες.
- Οι παρακείμενες περιοχές μιας κατάτμησης πρέπει να έχουν διαφορετικές τιμές από τις υπόλοιπες όσον αφορά το χαρακτηριστικό στο οποίο είναι ομοιόμορφες.

- Τα όρια κάθε τμήματος πρέπει να είναι ομαλά και ακριβή.

Η επίτευξη όλων αυτών των επιθυμητών ιδιοτήτων είναι δύσκολη καθώς οι ομοιόμορφες και ομοιογενείς περιοχές είναι γεμάτες από μικρές τρύπες και δεν έχουν ξεκάθαρα όρια. Στην περίπτωση που οι περιοχές δεν έχουν πολύ μεγάλες διαφορές, υπάρχει ο κίνδυνος να συγχωνευθούν περιοχές και να χαθούν τα όρια. Επιπλέον, μια περιοχή που το ανθρώπινο μάτι παρατηρεί ως ομοιογενή, μπορεί να μην είναι ομοιογενής κατά την άποψη ενός συστήματος κατάτμησης που χρησιμοποιεί χαρακτηριστικά γνωρίσματα χαμηλού επιπέδου και ενδεχομένως να απαιτείται υψηλότερου επιπέδου γνώση.

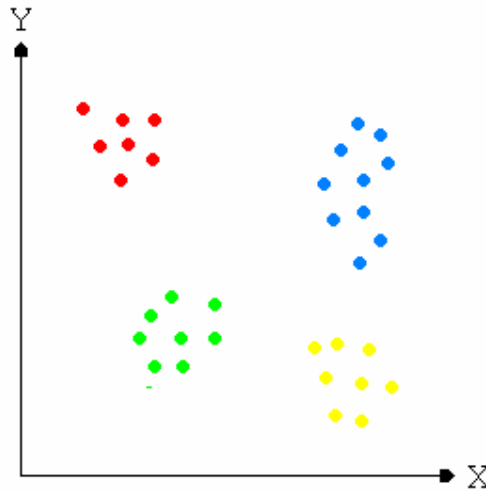
Υπάρχουν πολλοί αλγόριθμοι και τεχνικές κατάτμησης εικόνας. Οι πιο δημοφιλείς από αυτές είναι οι αλγόριθμοι ανάπτυξης περιοχών (region growing), οι αλγόριθμοι συστοιχιών (clustering), τεχνικές με βάση το ιστόγραμμα (histogram-based), αλγόριθμοι ανίχνευσης ακμών (edge detectors) κλπ.

Μέθοδοι Ανάπτυξης περιοχών (region growing methods)

Ένας καλλιεργητής περιοχών (region grower) αρχίζει από μια θέση στην εικόνα (συνήα η κορυφαία αριστερή γωνία) και προσπαθεί να αυξήσει κάθε περιοχή έως ότου τα pixels που συγκρίνονται να είναι πάρα πολύ ανάμοια για να προσθέσουν στην περιοχή. Συνήθως εκτελείται μια στατιστική δοκιμή για να αποφασίσει εάν αυτό συμβαίνει.

Μέθοδοι συστάδων (Clustering Methods)

Η μέθοδος clustering στην αναγνώριση προτύπων είναι η διαδικασία χωρίσματος ενός συνόλου διανυσμάτων σε υποσύνολα που ονομάζονται clusters. Παραδείγματος χάριν, εάν τα διανύσματα είναι ζευγάρια αριθμών όπως εμφανίζονται στο παρακάτω σχήμα, η κάθε συστάδα αποτελείται από τα σημεία που είναι πολύ κοντινά το ένα στο άλλο με βάση κάποιο κριτήριο.



Σχήμα 2.1 – Clustering Methods

Το γενικό πρόβλημα του clustering είναι να χωριστεί ένα σύνολο διανυσμάτων σε ομάδες που έχουν τις παρόμοιες τιμές. Στην ανάλυση εικόνας, τα διανύσματα αντιπροσωπεύουν τα pixels ή μερικές φορές μικρές περιοχές γύρω από τα pixels. Τα συστατικά αυτών των διανυσμάτων μπορούν να περιλάβουν τιμές έντασης, τιμές RGB και ιδιότητες χρώματος που προσδιορίζονται από αυτές, μετρήσεις υψής κλπ. Οποιοδήποτε χαρακτηριστικό γνώρισμα που μπορεί να συνδεθεί με ένα pixel μπορεί να χρησιμοποιηθεί στα pixels του κάθε cluster. Μόλις τα pixels ομαδοποιηθούν σε clusters που βασίζονται στις τιμές της μέτρησης, είναι εύκολο να βρεθούν οι συνδεδεμένες περιοχές που χρησιμοποιούν τα συνδεδεμένα.

Μία από τις βασικότερες τεχνικές Clustering είναι ο αλγόριθμος K-Means, όπου υπάρχουν K clusters C_1, C_2, \dots, C_K με τα αντίστοιχα μέσα m_1, m_2, \dots, m_K . Για να υπολογισθεί το πόσο κοντά βρίσκονται τα δεδομένα στα clusters όπου έχουν ανατεθεί, χρησιμοποιείται ο παρακάτω τύπος.

$$D = \sum_{k=1}^k \sum_{x_i \in C_k} \|x_i - m_k\|^2$$

Μία clustering διαδικασία ελάχιστων τετράγωνων μπορεί να συγκεντρώσει όλα τα πιθανά διαμερίσματα σε K clusters και να επιλέξει αυτό που ελαχιστοποιεί το D . Δεδομένου ότι αυτό είναι υπολογιστικά ανέφικτο, οι δημοφιλείς μέθοδοι είναι εκτιμήσεις. Ένα σημαντικό ζήτημα είναι εάν το K είναι ή όχι γνωστό εκ των

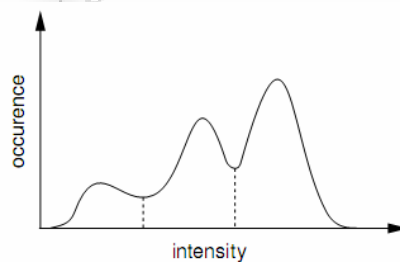
προτέρων. Πολλοί αλγόριθμοι δέχονται το K ως παράμετρο από το χρήστη. Άλλοι προσπαθούν να βρουν το καλύτερο K σύμφωνα με κάποιο κριτήριο, όπως την παραμονή της τιμής της διασποράς του κάθε cluster κάτω από μια συγκεκριμένη τιμή.

Μέθοδοι βασισμένες στο Ιστόγραμμα (Histogram-based methods)

Τεχνικές όπως οι δημιουργία συστάδων, απαιτούν περισσότερα από ένα πέρασματα των στοιχείων της εικόνας. Γι' αυτό το λόγο, συνήθως απαιτούν μεγάλο χρόνο υπολογισμού. Οι μέθοδοι ιστογραμμάτων, επειδή απαιτούν μόνο ένα πέρασμα των στοιχείων της εικόνας, απαιτούν λιγότερο χρόνο υπολογισμού.

Σε αυτήν την τεχνική, υπολογίζεται ένα ιστόγραμμα από όλα τα pixels της εικόνας και οι ακμές και οι κοιλάδες του ιστογράμματος χρησιμοποιούνται για να εντοπίσουν τις συστάδες στην εικόνα [1].

Οι προσεγγίσεις κατωφλίου, τέμνουν τις εικόνες με τη δημιουργία ενός δυαδικού χωρισμού των εντάσεων εικόνας. Μια διαδικασία κατωφλίου, προσπαθεί να καθορίσει μια τιμή έντασης, αποκαλούμενη κατώτατο όριο, το οποίο χωρίζει τις επιθυμητές κατηγορίες. Η κατάτμηση επιτυγχάνεται έπειτα από ομαδοποίηση όλων των pixels που έχουν την ένταση μεγαλύτερη από το κατώτατο όριο σε μια κατηγορία, και όλα τα υπόλοιπα pixels σε μια άλλη κατηγορία. Δύο πιθανά κατώτατα όρια παρουσιάζονται στις κοιλάδες του ιστογράμματος του παρακάτω σχήματος. Ο προσδιορισμός περισσότερων από μιας αξιών κατώτατων ορίων είναι μια διαδικασία αποκαλούμενη multithresholding [5].



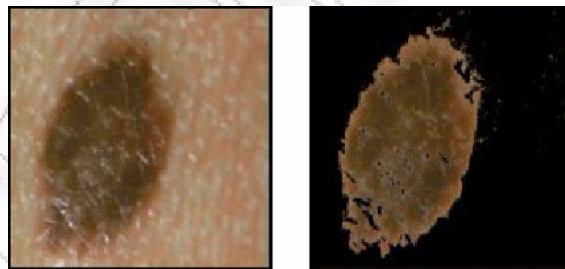
Σχήμα 2.2 – Ιστόγραμμα που παρουσιάζει τρεις κατηγορίες

Άλλες προσεγγίσεις

Καθώς το πρόβλημα της κατάτμησης εικόνας είναι ακόμα ανοικτό, πολλές νέες προσεγγίσεις παρατηρούνται. Ορισμένες από αυτές είναι εξολοκλήρου νέες, άλλες είναι συνδυασμός των παλαιότερων.

Σε μία προσπάθεια λύσης τέτοιου είδους προβλημάτων, υιοθετείται η εξελικτική στρατηγική των γενετικών αλγορίθμων για τη συγκέντρωση μικρών περιοχών στο διάστημα χαρακτηριστικών γνωρισμάτων του χρώματος. Η προσέγγιση χρησιμοποιεί τη μέθοδο k-means clustering στους γενετικούς αλγορίθμους, δηλαδή για την καθοδήγηση αυτού του τελευταίου εξελικτικού αλγορίθμου στην αναζήτησή της βέλτιστης κατανομής των δεδομένων. Σκοπός είναι να παρουσιαστεί η αποδοτικότητα των γενετικών αλγορίθμων στην αυτόματη και ανεπίβλεπτη κατάτμηση υφής [13].

Στις παρακάτω εικόνες παρουσιάζεται ένα παράδειγμα εφαρμογής της παραπάνω προσέγγισης. Στην πρώτη εικόνα παρουσιάζεται ανθρώπινος καρκίνος του δέρματος που ανιχνεύεται κυρίως από την ένταση περιμέτρου και χρώματος. Στη δεύτερη εικόνα παρουσιάζεται η απομονωμένη περιοχή.

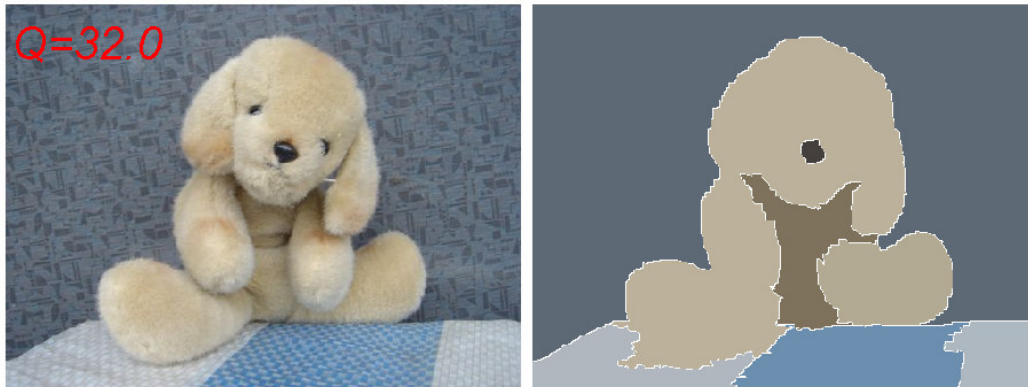


Εικόνα 2.3 – Παράδειγμα κατάτμησης εικόνας

Μία διαφορετική προσέγγιση παρουσίασαν οι Richard Nock και Frank Nielsen στην εργασία τους Statistical Region Merging (στατιστική συγχώνευση περιοχών). Στην εργασία αυτή παρουσιάζεται ένας συνδυασμός των αλγόριθμων και των στατιστικών των οποίων λάθος κατάτμησης είναι περιορισμένο και από ποσοτική και από ποιοτικές σκοπιά.

Αυτή η προσέγγιση οδηγεί σε έναν γρήγορο αλγόριθμο κατάτμησης, προσαρμοσμένος να επεξεργάζεται εικόνες στις εικόνες επεξεργασίας που περιγράφονται χρησιμοποιώντας τα πιο κοινά αριθμητικά διαστήματα ιδιοτήτων του pixel. Η απλότητα της προσέγγισης, την καθιστά απλή να τροποποιηθεί και να αντιμετωπίσει τη φορά του θορύβου, να επιτρέπει τον έλεγχο της κλίμακας κατάτμησης και να επεξεργάζεται τα μη συμβατικά δεδομένα όπως οι σφαιρικές εικόνες [11].

SRMj - Statistical Region Merging in Java by F. Nielsen and R. Nock



Εικόνα 2.4 – Παράδειγμα Αλγόριθμου Κατάτμησης

Στατιστικής Συγχώνευσης Περιοχών

2.1.3 Συνδυασμός των τεχνικών

Στο χώρο της ανάκτησης εικόνας βάσει περιεχομένου, χρησιμοποιούνται προσεγγίσεις όπου συνδυάζουν τις ήδη γνωστές τεχνικές. Το κρίσιμο σημείο κάθε φορά είναι το σύνολο των χαρακτηριστικών προς επεξεργασία. Η υφή συλλαμβάνει τη χωρική κατανομή των παραλλαγών έντασης φωτισμού (luminance variations). Το σχήμα αντιπροσωπεύει ένα συγκεκριμένο χαρακτηριστικό γνώρισμα ακμών που συσχετίζεται με το περίγραμμα αντικειμένου. Άλλα δομικά χαρακτηριστικά γνωρίσματα είναι η πυκνότητα των ακμών, η κατεύθυνση των ακμών κλπ.

Μία από τις ιδέες είναι η χρήση ενός συνόλου χαρακτηριστικών που ονομάζεται δομημένο (structural features) το οποίο έχει σαν χαρακτηριστικά την υφή και το σχήμα [15]. Ο αλγόριθμος ψάχνει τα μέτρα για το μήκος των ακμών, τη δομή των ακμών και την πολυπλοκότητα.

Εξαιτίας της αποτελεσματικότητάς του, ο αλγόριθμος του Canny εξακολουθεί να χρησιμοποιείται σε πολλές ερευνητικές εργασίες. Οι Jan Klíma και Tomas Skopal [9] επέλεξαν αυτή ως κύρια προσέγγιση για την ανίχνευση ακμών, δεδομένου ότι είναι κατά αποδεκτό τρόπο σταθερή και παραμετροποιήσιμη. Όμως σε πραγματικές εικόνες (φωτογραφίες), η ανίχνευση ακρών δεν μπορεί να εγγυηθεί ξεκάθαρα σχήματα, αλλά συνήθως υπάρχουν μεγάλα ποσά ψευδών ή ανεπιθύμητων ακρών. Για το λόγο αυτό είναι υποτιθέμενη μια επαναληπτική περικοπή ώστε να φροντίσει το μεγαλύτερο μέρος των ανεπιθύμητων ακμών στη διανυσματική έξοδο και επίσης γίνεται μέγιστη προσπάθεια για τη σύνδεση των γραμμών, την ανίχνευση γωνιών και την πολυγωνική εκτίμηση.

2.2 Εξαγωγή χαρακτηριστικών γνωρισμάτων από βίντεο

Ανάλυση των στοιχείων όπως το χρώμα και η υφή, που στοχεύουν στο χαρακτηρισμό των χαρακτηριστικών γνωρισμάτων στο χωρικό διάστημα, μπορούν να χρησιμοποιηθούν σε κάθε καρέ ενός βίντεο, με τον ίδιο ακριβώς τρόπο όπως και σε μία απλή εικόνα. Η εμπειρία που έχει αποκτηθεί από τις μελέτες πάνω στην ανάκτηση εικόνας, μπορεί να μεταφερθεί πλήρως σε ένα βίντεο. Επιπλέον όμως σε ένα βίντεο, είναι δυνατή η χρήση χρονικών και ακουστικών πληροφοριών για χαρακτηρισμό και την κατηγοριοποίηση.

Η χρονική διάσταση ενός αρχείου βίντεο περιέχει επιπλέον πληροφορίες. Η χρονική ανάλυση του βίντεο απαιτεί το χωρισμό του σε βασικά στοιχεία. Ο χωρισμός αυτός μπορεί να λειτουργήσει σε τέσσερα διαφορετικά επίπεδα[12].

- α) Frame level (επίπεδο καρέ): Κάθε καρέ επεξεργάζεται χωριστά.
- β) Shot-level (επίπεδο στιγμιότυπου): Ένα στιγμιότυπο είναι ένα σύνολο από συνεχόμενα καρέ που δημιουργούνται μέσω μιας συνεχούς καταγραφής μιας κάμερας. Ο χωρισμός του βίντεο σε στιγμιότυπα γενικά δεν αναφέρεται σε σημασιολογική ανάλυση. Χρησιμοποιούνται μόνο οι χρονικές πληροφορίες.
- γ) Scene-level (επίπεδο σκηνής): Μια σκηνή είναι ένα σύνολο από στιγμιότυπα που έχουν μια κοινή σημασιολογική σημασία.
- δ) Video-level (επίπεδο βίντεο): Το πλήρες βίντεο που αντιμετωπίζεται συνολικά.

Από τα τέσσερα επίπεδα, το βασικότερο κρίνεται πως είναι το επίπεδο στιγμιότυπου. Υπάρχουν τρεις τύποι ορίων στιγμιότυπου.

α) Cut (τομή): Ένα απότομο όριο μεταξύ των στιγμιότυπων. Αυτό υποδηλώνει μια κορυφή στη διαφορά μεταξύ των ιστογραμμάτων χρώματος ή κίνησης που αντιστοιχούν στα δύο καρέ που περιβάλλουν την τομή. Η ανίχνευση τομών επομένως συνίσταται στην ανίχνευση τέτοιων κορυφών. Η προσθήκη οποιασδήποτε μορφής χρονικής εξομάλυνσης μπορεί να βελτιώσει τη διαδικασία ανίχνευσης.

β) Dissolve (ξεθωρίασμα): Το περιεχόμενο των τελευταίων εικόνων του πρώτου στιγμιότυπου αναμιγνύεται με αυτό των πρώτων εικόνων του δεύτερου στιγμιότυπου. Το σημαντικό θέμα εδώ είναι να γίνει διάκριση μεταξύ των αποτελεσμάτων του ξεθωριάσματος και των αλλαγών που προκαλούνται από την κίνηση. Οι επενέργειες fade-in και fade-out είναι πρόσθετες περιπτώσεις ξεθωριάσματος όπου η πρώτη ή η δεύτερη σκηνή είναι αντίστοιχα ένα σκοτεινό πλαίσιο.

γ) Wipe (εξάλειψη): Οι εικόνες του δεύτερου στιγμιότυπου καλύπτουν αυτές του πρώτου στιγμιότυπου.

Παρόλο που η ανίχνευση τομών είναι σχετικά εύκολη, το ξεθωρίασμα και η εξάλειψη είναι δυσκολότερο να ανιχνευθούν. Μερικές αποδοτικές λύσεις που εκμεταλλεύονται τη συμπίεσμένη δομή των αρχείων MPEG έχουν προταθεί και βασίζονται στην εκτίμηση κίνησης και την κατάτμηση.

2.2.1 Χαρακτηριστικά γνωρίσματα κίνησης.

Ενώ το χρώμα η υφή και η οργάνωσή τους κατηγοριοποιούν το περιεχόμενο ενός βίντεο, κατά την επεξεργασία ενός τέτοιου αρχείου, είναι επίσης ουσιαστικό να υπολογιστεί η χρονική διάσταση. Τα χρονικά χαρακτηριστικά γνωρίσματα πρέπει να παρέχουν τις πληροφορίες σχετικά με τη χρονική οργάνωση ενός αρχείου βίντεο. Οι χρονικές πληροφορίες είναι γενικά μεταφρασμένες σε ένα χαρακτηριστικό κινήσεων. Η ανάλυση κινήσεων γίνεται ταιριάζοντας τα διαδοχικά πλαίσια μαζί. Και η αναζήτηση γίνεται μεταξύ των pixels δύο διαδοχικών πλαισίων.

Η στατιστική επιτρέπει το χαρακτηρισμό της σφαιρικής κίνησης (κυρίαρχη ή κίνηση φωτογραφικών μηχανών) και της κίνησης αντικειμένου. Χρησιμοποιώντας αυτή την πληροφορία, μπορεί να απομονωθεί η κίνηση ενός αντικειμένου και τέλος να εντοπιστεί το αντικείμενο αυτό.

2.2.2 Χαρακτηριστικά γνωρίσματα ήχου.

Όταν διατίθεται, η ροή ήχου που συνδέεται με ένα βίντεο μπορεί να αποτελέσει μεγάλη βοήθεια στην κατανόηση του. Χαρακτηριστικά, οι τεχνικές επεξεργασίας ήχου είναι βασισμένες στην ανάλυση της ενέργειας που περιλαμβάνεται στο ακουστικό σήμα. Το σήμα διαιρείται σε ηχητικά καρέ, που αντιστοιχούν σε λίγα χιλιοστά του δευτερολέπτου του σήματος. Στη συνέχεια χρησιμοποιούνται χαρακτηριστικά γνωρίσματα για το χαρακτηρισμό και την ταξινόμηση των ηχητικών καρέ.

Ένας στόχος συνίσταται στη διάκριση μεταξύ της ομιλίας και της μουσικής ή τον παρασιτικό θόρυβο στο ακουστικό σήμα. Υπάρχουν αλγόριθμοι που μπορούν να επιτύχουν αυτόν τον στόχο με μεγάλη ακρίβεια, βασισμένοι στο γεγονός ότι η ομιλία και η μουσική έχουν τις αρκετά διαφορετικές φασματικές κατανομές και χρονικό πρότυπο. Οι αλγόριθμοι μεταγραφής ομιλίας και κατάτμησης είναι πολύ προηγμένοι και επιτρέπουν την κατάτμηση ενός τηλεοπτικού εγγράφου όταν ο συγχρονισμένος ήχος και η εικόνα τέμνονται.

3

Ανάκτηση Πολυμέσων Βάσει Περιεχομένου.

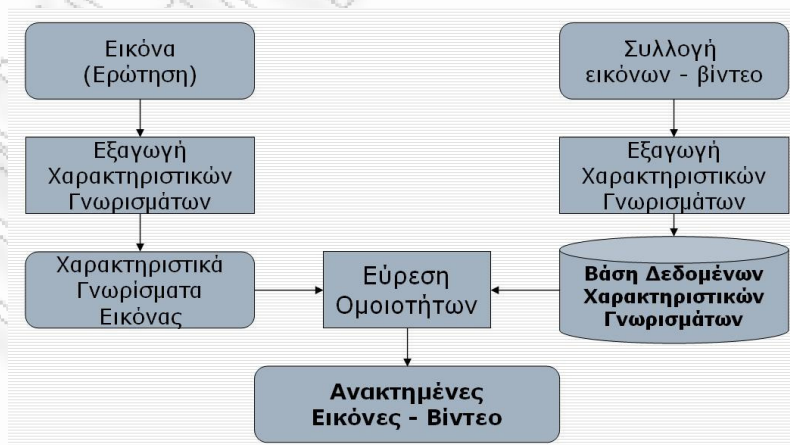
Η ανάκτηση εικόνας βάσει περιεχομένου (Content-Based Image Retrieval – CBIR) είναι το γνωστό πρόβλημα του τομέα της επιστήμης των υπολογιστών που λέγεται ψηφιακή όραση και αφορά στην αναζήτηση ψηφιακών εικόνων σε μεγάλες βάσεις δεδομένων. Η λέξη “περιεχόμενο”, χρησιμοποιείται για να δηλώσει την ανάλυση που απαιτείται να γίνει στα χαρακτηριστικά της εικόνας. Η λέξη χαρακτηριστικό μιας εικόνας μπορεί να αφορά τα χρώματά της, το σχήμα, την υφή (texture) ή οποιαδήποτε άλλη πληροφορία μπορεί να αποσπαστεί από μία ψηφιακή εικόνα.

Η λογική είναι κάθε φορά να εξετάζεται το περιεχόμενο κάθε εικόνας, όπου η κάθε εικόνα ταξινομείται σύμφωνα με την ομοιότητα της σε μια εικόνα ερώτησης (query image). Οι εικόνες που ανακτώνται είναι αυτές που παρουσιάζουν σημαντική ομοιότητα με την εικόνα ερώτησης.

Η διαρκής βελτίωση της επεξεργαστικής ικανότητας των υπολογιστών, σε συνδυασμό με τη ελάττωση του κόστους της μνήμης, οδήγησαν στο να γίνουν πραγματικότητα μεγάλες βάσεις δεδομένων εικόνων. Τα δεδομένα αυτά μπορεί να είναι εικόνες έργων τέχνης, δορυφορικές εικόνες, ιατρικές εικόνες και οποιαδήποτε άλλη συλλογή φωτογραφιών.

Η χρήση μίας τέτοιας βάσης δεδομένων εξαρτάται από την εφαρμογή που συνδέεται με τη βάση αλλά και από τις ανάγκες των χρηστών. Οι συλλέκτες έργων τέχνης μπορεί να επιθυμούν να βρουν ένα έργο ενός συγκεκριμένου καλλιτέχνη ή να ανακαλύψουν ποιος ζωγράφισε μια ιδιαίτερη εικόνα που έχουν δει. Οι φοιτητές ιατρικής που μελετούν την ανατομία ή γιατροί που ψάχνουν τις περιπτώσεις δειγμάτων μιας δεδομένης ασθένειας μπορούν να αναζητούν δεδομένα σε βάσεις ιατρικών εικόνων. Τα κριτήρια που μπορεί να χρησιμοποιήσει κάποιος για να πραγματοποιήσει μία αναζήτηση είναι απεριόριστα. Π.χ. ένας χρήστης μπορεί να ενδιαφέρεται να βρει εικόνες από ζώα (αλόγα, λιοντάρια ελέφαντες κλπ), κάποιος άλλος μπορεί να ψάχνει παραλίες, ενώ ένα τρίτο άτομο μπορεί να αναζητά μια αφηρημένη έννοια όπως η αγάπη.

Οι βάσεις δεδομένων εικόνων μπορούν να είναι τεράστιες, περιέχοντας εκατομμύρια εικόνες. Στις περισσότερες περιπτώσεις καταχωρούνται με λέξεις κλειδιά που πρέπει να αποφασιστούν για να εισαχθούν στο σύστημα βάσεων δεδομένων από ένα ανθρώπινο που αναλαμβάνει να τις κατηγοριοποιήσει. Εντούτοις, οι εικόνες μπορούν να ανακτηθούν σύμφωνα με το περιεχόμενό τους, όπου το περιεχόμενο μπορεί να αναφέρεται στην κατανομή του χρώματος, στην υφή, στο σχήμα κ.α. Ενώ η κατάσταση των αλγορίθμων κατάτμησης και αναγνώρισης είναι ακόμα πρωτόγονη, τα εμπορικά και ερευνητικά συστήματα έχουν κατασκευαστεί και είναι ήδη σε λειτουργία, συχνά διαθέσιμα στον παγκόσμιο ιστό. Στο σχήμα 3.1 παρουσιάζεται ένα γενικό μοντέλο των συστημάτων ανάκτησης εικόνας και βίντεο βάσει περιεχομένου.

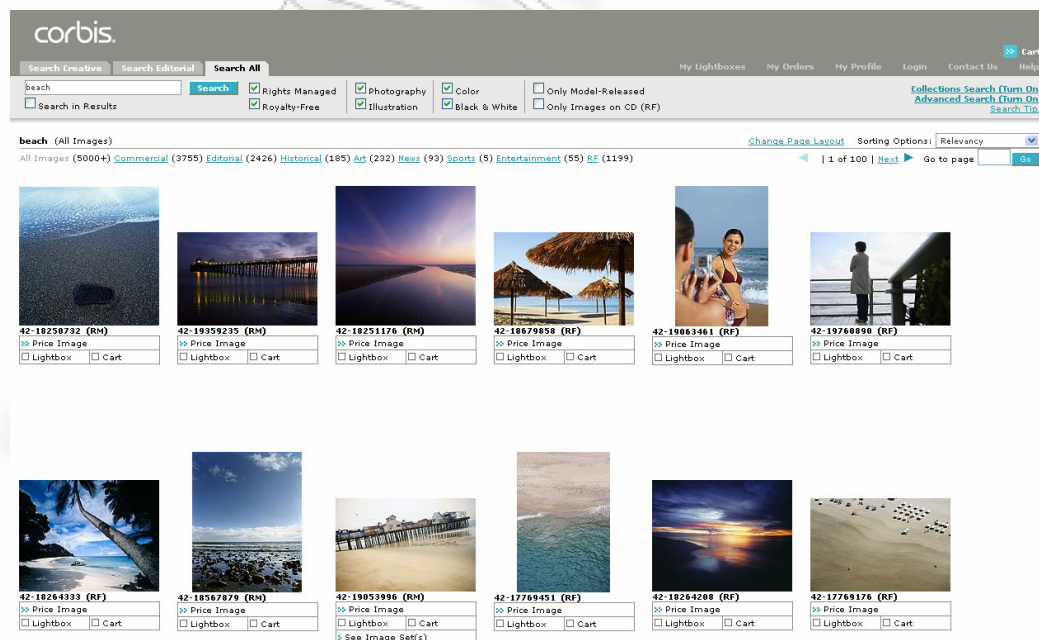


Σχήμα 3.1 – Γενικό μοντέλο συστημάτων CBIVR

3.1 Βάσεις Δεδομένων Εικόνων

Τα παραδείγματα βάσεων δεδομένων με εικόνες είναι πάρα πολλά. Η “ερώτηση με το περιεχόμενο της εικόνας” (Query by Image Content) της IBM είναι ένα ερευνητικό σύστημα που οδήγησε σε ένα εμπορικό προϊόν που αναπτύχθηκε από τη IBM [14]. Το QBIC ανακτά εικόνες βασισμένες στο οπτικό περιεχόμενο, συμπεριλαμβανομένων ιδιοτήτων όπως το ποσοστό χρώματος, τη διάταξη χρώματος και την υφή. Ανταγωνιστικό προϊόν ήταν η μηχανή αναζήτησης Virage, η οποία μπορεί να ανακτήσει εικόνες βασισμένες στο χρώμα, τη σύνθεση, την υφή, και τη δομή. Τέτοιου είδους μηχανές αναζήτησης εικόνων μπορούν να αναζητήσουν σε βάσεις δεδομένων που παρέχονται από άλλα συστήματα.

Μεγάλες ψηφιακές συλλογές τέχνης είναι δημιουργημένες και οργανωμένες σε πολλά μέρη του κόσμου. Εκτός από τις συλλογές τέχνης, υπάρχουν γενικές συλλογές εικόνων διαθέσιμες σε πελάτες που μπορούν να τις χρησιμοποιήσουν για το μάρκετινγκ των προϊόντων τους, ή την επεξήγηση των άρθρων τους. Ένα από τα μεγαλύτερα είναι το αρχείο Corbis (Corbis Archive), που περιέχει περισσότερες από 100 εκατομμύρια. Στις κατηγορίες των εικόνων συμπεριλαμβάνονται η ιστορία, η τέχνη, η ψυχαγωγία, η επιστήμη η βιομηχανία, τα ζώα κ.α. Το Corbis παρέχει ανάκτηση εικόνων με βάση τη λέξη κλειδί.



Εικόνα 3.1 – Βάσεις δεδομένων εικόνων

Εκτός από τα έργα τέχνης και τις φωτογραφίες, υπάρχουν επίσης επιστημονικές και ιατρικές συλλογές εικόνων όπως π.χ. συλλογές από ακτίνες X. Η NASA συλλέγει τεράστιες βάσεις δεδομένων εικόνων από τους δορυφόρους της και τις καθιστά διαθέσιμες δημόσια. Τέλος, ο παγκόσμιος ιστός είναι μια βάση δεδομένων που περιέχει εκτός από κείμενο έναν τεράστιο αριθμό εικόνων με τις μηχανές αναζήτησης να αναζητούν εικόνες στον Ιστό.

3.2 Ερωτήματα εικόνων

Σε τέτοιου μεγέθους βάσεις δεδομένων είναι ιδιαίτερα σημαντικό να υπάρχει κάποιος τρόπος ανάκτησης των εικόνων. Στα σχεσιακά συστήματα βάσεων δεδομένων, οι οντότητες μπορούν να ανακτηθούν βασισμένες στις τιμές των λεκτικών γνωρισμάτων. Οι εικόνες μπορούν να κατηγοριοποιηθούν σύμφωνα με αυτά τα γνωρίσματα, έτσι ώστε να μπορούν να ανακτηθούν γρήγορα όταν εκτελεστεί μια ερώτηση. Αυτός ο τύπος ερώτησης μπορεί να εκφραστεί με τη γλώσσα SQL.

Παραδείγματος χάριν, η ερώτηση

```
SELECT * FROM IMAGE_DB
WHERE CATEGORY = 'ANIMALS'
AND SPECIES = 'FELID';
```

θα έβρισκε και θα επέστρεφε όλες τις εικόνες που ανήκουν στην κατηγορία “animals” και έχουν όνομα “felid”. Προκειμένου να γίνει η αναζήτηση πιο συγκεκριμένη, θα πρέπει να αποθηκευτεί ένα περιγραφικό σύνολο λέξεων κλειδιών (keywords) για κάθε εικόνα. Η λέξη κλειδί είναι μια ιδιότητα που μπορεί να έχει περισσότερες από μία τιμές για κάθε εικόνα. Π.χ. η αναζήτηση για ένα τσίτα θα μπορούσε να έχει ως λέξεις κλειδιά τα “fast”, “speed”, “wild” και η ερώτηση SQL θα γινόταν

```
SELECT * FROM IMAGE_DB
WHERE CATEGORY = 'ANIMALS'
AND SPECIES = 'FELID'
AND (KEYWORD = 'FAST' OR KEYWORD = 'SPEED' OR KEYWORD = 'WILD');
```


Το πρόβλημα με αυτήν την προσέγγιση είναι ότι είναι περιοριστική. Η δημιουργία των λέξεων κλειδιών από ανθρώπους, εκτός από το ότι κοστίζει, αφήνει έξω όρους με βάση των οποίων κάποιοι χρήστες μπορεί να αναζητούσαν μία συγκεκριμένη εικόνα.

Μία πολύ καλή ιδέα για την ανάκτηση εικόνων από μία βάση δεδομένων, θα ήταν να μπορούσε ο χρήστης να δώσει στο σύστημα μία εικόνα, ή να τη ζωγραφίσει στην οθόνη ή να σχεδιάσει το περίγραμμα ενός αντικειμένου. Αυτός είναι και ο στόχος όλων των συστημάτων ανάκτησης εικόνας βάση περιεχομένου. Το κάθε σύστημα έχει δικό του τρόπο να ορίζει το πώς θα γίνει μία ερώτηση, πώς καθορίζεται η ομοιότητα μεταξύ της ερώτησης και των εικόνων στη βάση δεδομένων και τελικά την επιλογή των εικόνων που επιστρέφονται.

Σε πολλές περιπτώσεις, η ερώτηση είναι μια ψηφιακή εικόνα που συγκρίνεται με τις εικόνες στη βάση δεδομένων σύμφωνα με ένα μέτρο απόστασης εικόνας (image distance measure). Όταν η επιστρεφόμενη τιμή της απόστασης είναι μηδέν η εικόνα ταιριάζει ακριβώς με την ερώτηση. Οι τιμές μεγαλύτερες του μηδέν δείχνουν τους διάφορους βαθμούς ομοιότητας στην ερώτηση. Οι μηχανές αναζήτησης εικόνας συνήθως επιστρέφουν το σύνολο των αποτελεσμάτων ταξινομημένο με βάση την απόστασή τους από το ερώτημα.

3.3 Υπολογισμός της ομοιότητας

Υπάρχουν τρία κύρια χαρακτηριστικά βάσει των οποίων μπορεί να υπολογιστεί η ομοιότητα μεταξύ ενός ερωτήματος και των εικόνων – αποτελεσμάτων. Αυτά είναι το χρώμα, η υφή, και το σχήμα.

3.3.1 Μέτρα ομοιότητας χρώματος

Τα μέτρα ομοιότητας χρώματος συνήθως είναι απλά. Συγκρίνουν το χρωματικό περιεχόμενο μίας εικόνας με το χρωματικό περιεχόμενο μιας δεύτερης εικόνας ή μιας προδιαγραφής ερώτησης. Μια τεχνική είναι το ταίριασμα ιστογραμμάτων χρώματος (color histogram). Ο χρήστης παρέχει στο σύστημα ένα δείγμα εικόνας και ζητάει να του επιστραφούν όλες οι εικόνες των οποίων η χρωματική απόσταση του ιστογράμματος είναι μικρή. Οι χρωματικές αποστάσεις του ιστογράμματος πρέπει να περιλαμβάνουν κάποιες μετρήσεις για το πόσο όμοια είναι δύο διαφορετικά χρώματα.

Ένα άλλο πιθανό μέτρο απόστασης είναι το σχεδιάγραμμα χρώματος (color layout). Ο χρήστης μπορεί να αρχίσει με ένα κενό πλέγμα που αντιπροσωπεύει την

ερώτηση και να επιλέξει τα χρώματα για κάθε ένα από τα τετράγωνα του πλέγματος από έναν πίνακα. Η αναπαράσταση του χρώματος σε ένα τετράγωνο του πλέγματος μπορεί να είναι πολύ απλή ή περίπλοκη. Κατάλληλα χαρακτηριστικά αναπαράστασης είναι το μέσο χρώμα στο τετράγωνο πλέγματος, η μέση και σταθερή απόκλιση του χρώματος, και το ιστόγραμμα του χρώματος.

3.3.2 Μέτρα ομοιότητας υφής

Τα μέτρα ομοιότητας υφής είναι πιο σύνθετα από τα μέτρα ομοιότητας χρώματος. Μια εικόνα που έχει παρόμοια υφή σε μια ερώτηση πρέπει να έχει τις ίδιες χωρικές ρυθμίσεις χρωμάτων, αλλά όχι απαραίτητα τα ίδια χρώματα. Τα μέτρα απόστασης υφής έχουν δύο πτυχές. α) την αναπαράσταση της υφής, και β) τον καθορισμό της ομοιότητας όσον αφορά εκείνη την αναπαράσταση.

Η πιο συνηθισμένη αναπαράσταση υφής είναι ένα διάνυσμα περιγραφής (texture description vector) που είναι ένα διάνυσμα αριθμών που συνοψίζει την υφή σε μια δεδομένη εικόνα ή μια περιοχή εικόνας. Ενώ ένα διάνυσμα περιγραφής υφής μπορεί να χρησιμοποιηθεί για να συνοψίσει τη υφή σε μια ολόκληρη εικόνα, αυτό είναι αποτελεσματικό μόνο για την περιγραφή εικόνων ενιαίας υφής. Για πιο γενικές εικόνες, τα διανύσματα περιγραφής της υφής υπολογίζονται σε κάθε pixel για μια μικρή (15 x 15) γειτονική περιοχή για εκείνο το pixel. Κατόπιν τα pixels ομαδοποιούνται από έναν αλγόριθμο clustering που ορίζει μια μοναδική ετικέτα σε κάθε μια διαφορετική κατηγορία υφής που βρίσκει.

Δεδομένου ότι τα pixel προσδιορίζουν ένα διάνυσμα περιγραφής υφής και χαρακτηρίζονται πως ανήκουν σε μια κατηγορία, διάφορες αποστάσεις υφής μπορούν να καθοριστούν. Η απλούστερη απόσταση υφής είναι η “pick and click”. Ο χρήστης επιλέγει μια υφή κάνοντας click πάνω σε ένα pixel μίας περιοχής υφής της εικόνας – ερώτησης ή επιλέγοντας την υφή από ένα σύνολο επιλογών. Η επιλεγμένη υφή αναπαριστάται από το διάνυσμα περιγραφής της, το οποίο συγκρίνεται με τα διανύσματα περιγραφής υφής που συνδέονται με τη βάση δεδομένων. Το μέτρο απόστασης καθορίζεται από τον τύπο

$$D_{\text{pick_and_click}}(I, Q) = \min_{i \in I} \|T(i) - T(Q)\|^2$$

όπου το $T(i)$ είναι το διάνυσμα περιγραφής υφής στο pixel I της εικόνας I και $T(Q)$ είναι το διάνυσμα περιγραφής υφής στο επιλεγμένο pixel ή για την επιλεγμένη κατηγορία υφής της ερώτησης. Αν και αυτό μοιάζει με το να είναι υπολογιστικά

κοπιαστικό, το μεγαλύτερο μέρος του υπολογισμού μπορεί να αποφευχθεί με την αναπαράσταση μιας βάσης δεδομένων εικόνας από έναν κατάλογο κατηγοριών υφής όπως καθορίζεται από τη διαδικασία clustering. Για κάθε βάση δεδομένων εικόνας, η ερώτηση διανυσματικής περιγραφής υφής χρειάζεται να συγκριθεί μόνο με τα διανύσματα περιγραφής υφής του καταλόγου της. Το ευρετήριο μπορεί να παρέχει ακόμα και γρηγορότερη ανάκτηση.

Η παραπάνω μέθοδος απαιτεί από το χρήστη να επιλέξει μία υφή και δεν μπορεί να λειτουργήσει αυτόματα σε μια εικόνα – ερώτηση. Ένα γενικότερο μέτρο υφής είναι μια γενίκευση των μέτρων πλέγματος που συζητούνται από το χρώμα στη υφή. Ένα πλέγμα τοποθετείται πάνω από την ερώτηση – εικόνα και ένα διάνυσμα περιγραφής υφής υπολογίζεται για κάθε τετράγωνο πλέγματος. Η ίδια διαδικασία εφαρμόζεται στην εικόνα της βάσης δεδομένων. Οι αποστάσεις των ιστογραμμάτων υφής μπορούν να καθοριστούν με τρόπο παρόμοιο των ιστογραμμάτων χρώματος. Για κάθε κατηγορία υφής, το ιστόγραμμα καθορίζει τον αριθμό των pixels των οποίων το διάνυσμα περιγραφής υφής περιέρχεται σε εκείνη την κατηγορία.

3.3.3 Μέτρα ομοιότητας σχήματος

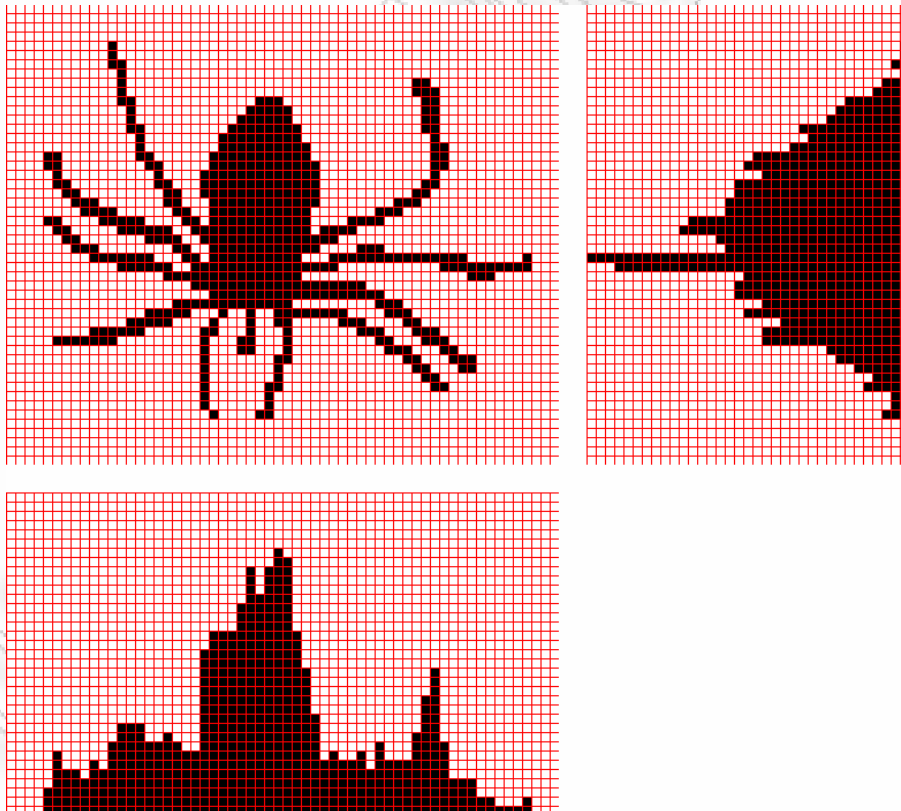
Το χρώμα και η υφή είναι ιδιότητες της εικόνας. Μέτρα απόστασης βασισμένα σε αυτά τα μεγέθη προσπαθούν να καθορίσουν εάν μια δεδομένη εικόνα έχει ένα καθορισμένο χρώμα ή μια καθορισμένη υφή και εάν εμφανίζεται στην ίδια κατά προσέγγιση θέση όπως στην εικόνα – ερώτηση. Το σχήμα δεν είναι ιδιότητα της εικόνας, αλλά αναφέρεται σε μια συγκεκριμένη περιοχή μιας εικόνας, και απαιτεί κάποια διαδικασία προσδιορισμού περιοχών να προηγηθεί του μέτρου ομοιότητας του σχήματος. Σε πολλές περιπτώσεις αυτό πρέπει να γίνει με τρόπο μη αυτοματοποιημένο, αλλά η αυτοματοποιημένη κατάτμηση είναι δυνατή σε αρκετές περιπτώσεις. Η κατάτμηση είναι ακόμα ένα σοβαρό πρόβλημα προς επίλυση για την ανάκτηση με βάση τη σχηματική πληροφορία .

Στα πλαίσια της ανάκτησης εικόνας, απαιτείται οι μέθοδοι που χρησιμοποιούνται να μπορούν γρήγορα να αποφασίσουν το ποσοστό ομοιότητας μεταξύ του σχήματος του αντικειμένου μίας εικόνας και του σχήματος της ερώτησης. Συχνά, απαιτείται οι τεχνικές ταιριάσματος των σχημάτων είναι σταθερές ως προς τη μετατόπιση και το μέγεθος. Επίσης πολλές φορές απαιτείται περιστροφική σταθερότητα, έτσι ώστε ένα αντικείμενο να μπορεί να προσδιοριστεί ακόμη και αν έχει διαφορετικό

προσανατολισμό. Αυτό βέβαια δεν είναι πάντοτε απαραίτητο αφού υπάρχουν πολλές εικόνες στις οποίες τα αντικείμενα εμφανίζονται συνήθως στο σωστό προσανατολισμό (π.χ. κτήρια, δέντρα κλπ).’

Ιστογράμμα σχήματος.

Επειδή οι αποστάσεις ιστογραμμάτων είναι γρήγορο και εύκολο να υπολογιστούν και χρησιμοποιούνται και για το ταίριασμα χρώματος και για το ταίριασμα υφής, είναι μια καλή ιδέα να χρησιμοποιηθούν και στη σύγκριση σχημάτων. Το κύριο πρόβλημα είναι να καθοριστεί η μεταβλητή στην οποία το ιστογράμμα είναι καθορισμένο. Ας θεωρήσουμε μια περιοχή από “1” σε μία δυαδική εικόνα όπου τα υπόλοιπα pixels είναι “0”. Ένα είδος ταιριάσματος ιστογραμμάτων είναι το ταίριασμα προβολής (projection matching) που ταιριάζει τις οριζόντιες και κάθετες προβολές του σχήματος.



Σχήμα 3.2 – Παράδειγμα ιστογράμματος σχήματος

Το σχήμα θα έχει i γραμμές και j στήλες. Κάθε γραμμή και κάθε στήλη γίνεται ένα δοχείο στο ιστόγραμμα. Αυτό που αποθηκεύεται σε ένα δοχείο είναι ο αριθμός των “1” pixels που εμφανίζονται σε εκείνη τη σειρά ή τη στήλη. Αυτό οδηγεί σε ένα ιστόγραμμα $i + j$ δοχείων που είναι χρήσιμο μόνο εάν το σχήμα έχει πάντα το ίδιο μέγεθος. Για να γίνει το ταίριασμα της προβολής ανεξάρτητο μεγέθους, ο αριθμός δοχείων σειρών και ο αριθμός δοχείων στηλών μπορούν να καθοριστούν. Ορίζοντας τα δοχεία από την πάνω αριστερή ως την κάτω δεξιά άκρη, μπορεί να επιτευχθεί σταθερότητα μετασχηματισμού. Υπάρχουν ακόμα διάφορες τεχνικές που μπορούν να κάνουν το ταίριασμα προβολής αμετάβλητο και κατά την περιστροφή. Στο παρακάτω σχήμα (σχήμα 3.2) παρουσιάζονται τα δύο ιστογράμματα που προκύπτουν από μία binary εικόνα μίας αράχνης.

Ταίριασμα ορίων.

Οι αλγόριθμοι ταιριάσματος ορίου απαιτούν την εξαγωγή και αναπαράσταση των ορίων του σχήματος – ερώτησης και της μορφής εικόνας. Το όριο μπορεί να αναπαρασταθεί ως μία ακολουθία από pixels ή μπορεί να προσεγγιστεί από ένα πολύγωνο. Για την ακολουθία των pixels, μια κλασσική προσέγγιση ταιριάσματος χρησιμοποιεί τους περιγραφείς Fourier για να συγκρίνει δύο σχήματα.

Ταίριασμα σκίτσων.

Σε γενικές γραμμές, τα συστήματα ταιριάσματος σκίτσων επιτρέπουν στο χρήστη να εισάγει στο σύστημα ένα σκίτσο και στη συνέχεια αυτό του επιστρέφει εικόνες των οποίων οι ακμές ταιριάζουν με το σχήμα. Οι A. Chalechale, G. Naghdy, και A. Mertins παρουσίασαν μία προσέγγιση εξαγωγής χαρακτηριστικών γνωρισμάτων, με σκοπό το ταίριασμα δύο εικόνων, βασισμένη στο γωνιακό χωρισμό [2]. Οι εικόνες λαμβάνονται από την εικόνα πρότυπο και από την εικόνα ερώτησης. Η γωνιακός-χωρική κατανομή των pixels της αρχικής εικόνας υιοθετείται ως βασική έννοια για την εξαγωγή χαρακτηριστικών γνωρισμάτων με χρήση του μετασχηματισμού Φουριέ. Τα εξαγόμενα χαρακτηριστικά γνωρίσματα είναι αμετάβλητα ως προς την κλίμακα και την περιστροφή και ισχυρά απέναντι στο μετασχηματισμό.

Το σύστημα αυτό βρίσκει εφαρμογή στην ανάκτηση εικόνας με βάση το σχήμα. Δίνει τη δυνατότητα στο χρήστη να ανακτά παρόμοιες εικόνες από τη βάση δεδομένων ή να ταξινομεί την ποιότητα του σκίτσου απέναντι σε κάποιο πρότυπο,

π.χ. την αρχική εικόνα. Η σημαντικότερη συμβολή του συστήματος είναι η ανάκτηση που πραγματοποιεί βασισμένη στο σχήμα (SBIR) η οποία δεν χρησιμοποιεί τεχνικές κατάτμησης, με αμετάβλητες ιδιότητες κλίμακας και περιστροφής. Ο συγκεκριμένος αλγόριθμος εξετάστηκε μαζί με άλλες έξι γνωστές από τη βιβλιογραφία προσεγγίσεις χρησιμοποιώντας ένα έργο και μια βάση δεδομένων εικόνων. Ο στόχος ήταν να παρουσιάσει ο βαθμός σταθερότητας της περιστροφής και της κλίμακας χωριστά. Πειραματικά αποτελέσματα επιβεβαίωσαν την ανωτερότητα της προτεινόμενης μεθόδου χρησιμοποιώντας το τυποποιημένο μέτρο αξιολόγησης απόδοσης της ανάκτησης MPEG-7

3.3.4 Σχεσιακά μέτρα ομοιότητας

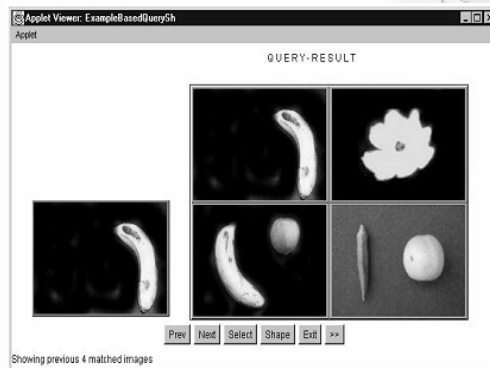
Μέτρα απόστασης όπως το χρώμα, η σύσταση, και το σχήμα, συνήθως δεν καλύπτουν τις απαιτήσεις των μέσων χρηστών, οι οποίοι τις περισσότερες φορές αναζητήσουν εικόνες που περιέχουν συγκεκριμένα αντικείμενα όπως οι άνθρωποι ή τα αλόγα ή μπορεί να ζητούν είναι αφηρημένες έννοιες, όπως η ευτυχία ή η αγάπη. Συστήματα που προσφέρουν αναγνώριση αντικειμένων ψάχνουν αντικείμενα όπως ανθρώπινα πρόσωπα, τα ανθρώπινα σώματα, ζώα κλπ.

Η εύρεση προσώπου είναι μία σημαντική τεχνική, αφού μας επιτρέπει να ψάχνουμε σε εικόνες τον περιορισμό των ανθρώπων. Δυσκολίες που παρουσιάζονται είναι ότι τα πρόσωπα βρίσκονται σε ποικίλα μεγέθη, οι θέσεις τους σε μια εικόνα μπορεί να διαφέρει ή να βρίσκεται σε άλλη άποψη, και επίσης υπάρχει ποικιλία χρωμάτων. Ένας άλλος τρόπος εύρεσης αντικειμένων είναι η εύρεση περιοχών σε εικόνες που το χρώμα και η υφή συνδέονται συνήθως με το ως προς αναζήτηση αντικείμενο. Μια από τις πρώτες προσπάθειες έγινε από τους Fleck, Forsyth, και Bregler με σκοπό την εύρεση εικόνων γυμνών ανθρώπων, η οποία είχε σκοπό να φιλτράρει και να εμποδίσει την πορνογραφία από τα αποτελέσματα ερωτημάτων [10]. Η μέθοδος αυτή έχει δύο κύρια βήματα: α) εύρεση μεγάλων περιοχών με πιθανή σάρκας και β) ομαδοποίηση αυτών των περιοχών για την εύρεση πιθανών ανθρώπινων σωμάτων.

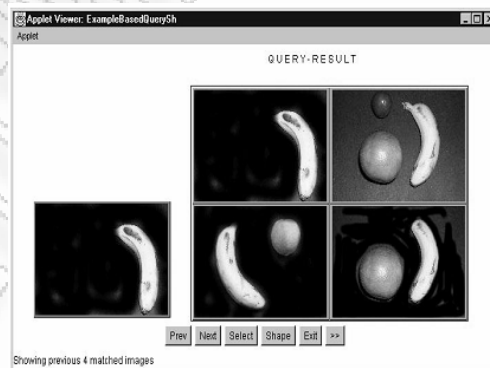
3.4 Συστήματα ανάκτησης

Η περιγραφή του σχήματος, είναι ένα σημαντικό ζήτημα τόσο στην αναγνώριση αντικειμένων, όσο και στην ταξινόμηση. Πολλές τεχνικές, συμπεριλαμβανομένου του κώδικα αλυσίδων (chain code), οι προσεγγίσεις πολυγώνων, η κυρτότητα, οι περιγραφείς Fourier κ.α. έχουν προταθεί και χρησιμοποιηθεί σε αρκετές εφαρμογές.

Σε μία προσέγγισή των Prasad, Gupta και Biswas [3], το πρώτο βήμα είναι η καταχώριση εικόνων, βασισμένες στις κυρίαρχες περιοχές χρώματος. Οι περιοχές των εικόνων που λαμβάνονται μετά την κατάτμηση και την καταχώριση χρησιμοποιούνται σαν εισαγωγή στο κομμάτι που αφορά το σχήμα. Στη συνέχεια μία αναπαράσταση σχήματος βασισμένη σε περιοχές χρησιμοποιείται για τον υπολογισμό των χαρακτηριστικών του σχήματος. Στις εικόνες 3.2.1 και 3.2.2 παρουσιάζονται δύο παραδείγματα του συστήματος αυτού. Στην εικόνα 3.2.1 έχουμε ανάκτηση εικόνας βασισμένη στις περιοχές χρώματος, ενώ στη δεύτερη εικόνα (3.2.2), η ανάκτηση είναι βασισμένη στο σχήμα.



Εικόνα 3.2.1



Εικόνα 3.2.2

3.5 Βάσεις Δεδομένων Βίντεο

Εκτός από τις βάσεις δεδομένων εικόνων, έχουν δημιουργηθεί πλέον και τεράστιες βάσεις δεδομένων που περιέχουν συλλογές από βίντεο. Μέρα με τη μέρα εμφανίζονται όλο και περισσότερες ιστοσελίδες οι οποίες δίνουν τη δυνατότητα σε χρήστες να έχουν πρόσβαση σε εκατομμύρια βίντεο.

The screenshot shows the YouTube homepage with the search bar containing 'The Prodigy'. The search results are displayed in a grid format. The first video is 'The Prodigy - Warriors Dance - Full Live Version' with 57,257 views and a 5-star rating. The second video is 'The Prodigy - Unstoppable - Full Live Version' with 15,763 views and a 5-star rating. The interface includes navigation tabs for Home, Videos, Channels, and Community, and a search bar with an 'Upload' button. The page also features a 'Sign Up' link and a 'QuickList (0)' indicator.

Εικόνα 3.3 – Βάσεις Δεδομένων Βίντεο

Ένα παράδειγμα ενός τέτοιου είδους συστήματος είναι το YouTube. Πρόκειται για μία ιστοσελίδα που δίνει σε χρήστες του Διαδικτύου τη δυνατότητα να αναζητούν τα βίντεο που επιθυμούν να δουν αλλά και να ανεβάζουν τα δικά τους. Σύμφωνα με έρευνα που πραγματοποιήθηκε το 2006 εκτιμήθηκε ότι χρήστες του YouTube παρακολουθούν περισσότερα από 100 εκατομμύρια βίντεο την ημέρα.

3.6 Ερωτήματα Βίντεο

Στην ανάκτηση εικόνας βάση περιεχομένου, ένα σύστημα ερώτησης βάση παραδείγματος είναι σχετικά διαισθητικό δεδομένου ότι καλύπτει περιπτώσεις που θα ήταν δύσκολο να λυθούν με χρήση ερωτήσεων κειμένου. Ωστόσο για τα βίντεο, τα πράγματα δεν είναι τόσο απλά δεδομένου ότι μία ερώτηση-παράδειγμα θα απαιτούσε από το χρήστη να έχει ένα δείγμα βίντεο. Ένα σοβαρό πρόβλημα είναι το μέγεθος του εύρους αναζήτησης που προκαλείται από τις χρονικές πληροφορίες. Προκειμένου αυτό να ελαττωθεί, διαφορές προσεγγίσεις υιοθετούνται και κάθε μία από αυτές έχει πλεονεκτήματα και μειονεκτήματα.

Εξαιτίας της δυσκολίας που αντιμετωπίζεται στην περιγραφή του περιεχομένου ενός βίντεο, δεν υπάρχουν πολλά συστήματα που χρησιμοποιούν τεχνικές αντίστοιχες με αυτές της ανάκτησης εικόνας (χαρακτηριστικά γνωρίσματα όπως το χρώμα και η

υφή). Μία άλλη προσέγγιση είναι η προσπάθεια αξιοποίησης της κίνησης. Η κίνηση φαίνεται να είναι ο μόνος τρόπος αναπαράστασης των χρονικών πληροφοριών που περιλαμβάνονται σε ένα βίντεο. Μία ερώτηση βασισμένη στην κίνηση είναι επομένως ένα ελκυστικό χαρακτηριστικό γνώρισμα για αναζήτηση βίντεο. Το πρόβλημα σε αυτή την περίπτωση είναι ο τρόπος με τον οποίο μπορεί να διατυπωθεί μια ερώτηση.

Σε μία εργασία τους, οι Edoardo Ardizzone, και Marco La Cascia περιγράφουν ένα σύστημα που επιτρέπει την πραγματοποίηση ερωτήσεων βάσει περιεχομένου, σε βάσεις δεδομένων με βίντεο [6]. Το σύστημα δεν απαιτεί καμία ενέργεια από το χρήστη. Χωρίζει αυτόματα ένα βίντεο σε μια ακολουθία στιγμιότυπων, εξάγει μερικά αντιπροσωπευτικά καρέ από κάθε στιγμιότυπο και υπολογίζει έναν περιγραφέα για κάθε καρέ βασισμένο στο χρώμα, τη σύσταση και την κίνηση. Το σύστημα επίσης δίνει τη δυνατότητα εκτέλεσης ερωτημάτων βασισμένα σε περισσότερα από ένα χαρακτηριστικά γνωρίσματα.

3.7 Υπολογισμός της ομοιότητας σε βίντεο

Τα συστήματα ανάκτησης με βάση το βίντεο (CBVR), αποτελούν μία φυσική εξέλιξη των συστημάτων ανάκτηση με βάση την εικόνα και τον ήχο. Εντούτοις, υπάρχουν διάφοροι παράγοντες που αγνοούνται κατά την εξέταση των εικόνων, αλλά μπορούν να εξεταστούν κατά την εξέταση βίντεο. Αυτοί οι παράγοντες συσχετίζονται κυρίως με τη χρονική διαθέσιμη πληροφορία από ένα βίντεο και την ύπαρξη ήχου.

Η χρονική πληροφορία περιλαμβάνει την έννοια της κίνησης για τα αντικείμενα που βρίσκονται στο βίντεο. Σε αντίθεση με τα συστήματα ανάκτηση εικόνας, που το αντικείμενο της αναζήτησης είναι μία ομάδα κατηγοριοποιημένων εικόνων, στην ανάκτηση βίντεο μπορεί το στοιχείο της αναζήτησης να είναι η συμπεριφορά ενός αντικειμένου στο βίντεο. Δύο βίντεο μπορεί να περιέχουν τα ίδια αντικείμενα αλλά να μη σχετίζονται ιδιαίτερα στο πλαίσιο μιας αναζήτησης. Ένα άλλο θέμα στην ανάκτηση βίντεο είναι η πολυπλοκότητα των συστημάτων ερωτήσεων. Τα ανεπτυγμένα συστήματα ανάκτησης επιτρέπουν ευελιξία στο χρήστη ώστε να μπορεί να διευκρινίσει τις παραμέτρους της ερώτησής του. Συστήματα ερώτησης-από-παράδειγμα που χρησιμοποιούνται χαρακτηριστικά σε συστήματα ανάκτησης εικόνας, απαιτούν από το χρήστη να παρουσιάσει στο σύστημα ένας ή περισσότερα

στοιχεία παρόμοια με αυτό που ψάχνει. Αυτή η διαδικασία ερώτησης είναι αρκετά πιο περίπλοκο να προσαρμοστεί στην ανάκτηση βίντεο.

Τα συστήματα ανάκτησης βίντεο είναι αρκετά πρόσφατα και δεν έχουν ακόμη αποσαφηνιστεί όλες οι λειτουργίες και οι εφαρμογές που μπορούν να είναι χρήσιμα. Εξ ορισμού, ένα σύστημα CBVR στοχεύει στην ενίσχυση ενός χειριστή (χρήστη) να ανακτήσει μια ακολουθία βίντεο (στόχος) μέσα σε μια ενδεχομένως μεγάλη βάση δεδομένων [12]. Τρεις σημαντικές περιπτώσεις μπορούν να ξεχωρίσουν. Μερικά παραδείγματα χρήσης τέτοιου είδους συστημάτων μπορεί να είναι τα ακόλουθα:

- Ραδιοτηλεοπτική αναμετάδοση ειδήσεων. Η ανάγκη για τη συλλογή όλων των απαραίτητων αρχείων, οδηγεί σε έναν μεγάλο όγκο σύντομων βίντεο. Αυτή η ιδιομορφία των ειδήσεων καθιστά ένα αυτοματοποιημένο σύστημα ανάκτησης πολύ ελκυστικό για την ανάκτηση πηγών με ένα συγκεκριμένο θέμα. Επίσης αξίζει να σημειωθεί ότι ο ήχος που συνδέονται με τέτοιου είδους βίντεο καθώς και οι τίτλοι ειδήσεων αποτελούν σημαντικά δεδομένα για την αυτοματοποιημένη διαδικασία ανάκτησης.
- Μουσικά video clip. Ο τεράστιος όγκος από αυτού του είδους βίντεο, οδηγεί στο συμπέρασμα πως θα ήταν ιδιαίτερα χρήσιμο ένα εργαλείο αυτόματης ανάκτησης. Τα χαρακτηριστικά τέτοιων εγγράφων συνήθως είναι δύσκολο να εκφραστούν βασισμένα στους σχολιασμούς λέξεων που αποθηκεύονται σε μια βάση δεδομένων. Για αυτόν τον τύπο συγκεκριμένων εφαρμογών, θα μπορούσε να ήταν χρήσιμη η ανάκτηση βάσει κάποιου βήματος χορού που περιγράφεται π.χ. από ένα σκίτσο.
- Εξ' αποστάσεως εκπαίδευση.
- Ιατρικές εφαρμογές.
- Διαφήμιση.

Σημαντικά θέματα για την εξασφάλιση της δυνατότητας χρήσης και λειτουργίας τέτοιων συστημάτων είναι:

- Η ύπαρξη μιας συμπαγούς και ολοκληρωμένης ακολουθίας βίντεο.
- Η δυνατότητα να παρέχονται διαφορετικές στρατηγικές αναζήτησης που προσαρμόζονται στον τύπο αναζήτησης που επιθυμεί να κάνει ο χρήστης.

4

Περιγραφή συστήματος.

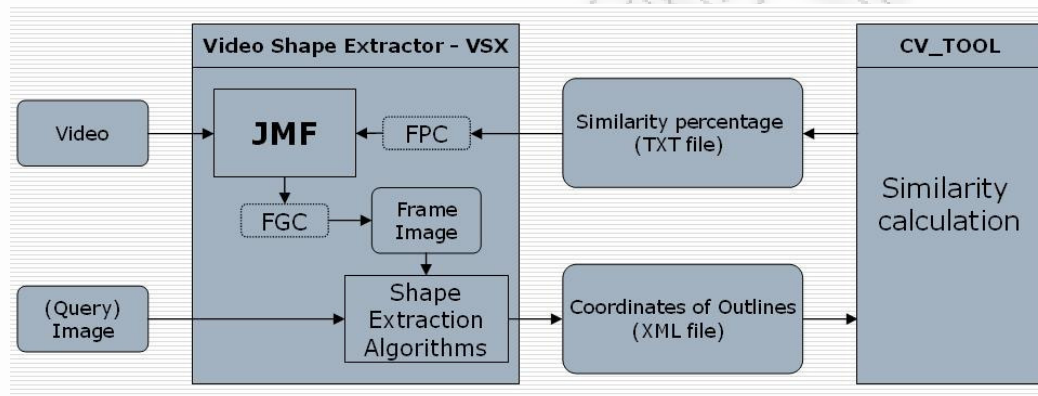
Για την ανάπτυξη του συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java (JDK 1.6.0). Για την αναπαραγωγή και την επεξεργασία των βίντεο χρησιμοποιήθηκε το πακέτο Java Media Framework (JMF2.1.1e) της Sun Microsystems. Η ανάπτυξη έγινε στο περιβάλλον NetBeans IDE 6.0.

Η εφαρμογή που αναπτύχθηκε αποτελείται από τρία μέρη. Το πρώτο μέρος είναι μία σειρά αλγορίθμων – βημάτων που εκτελούνται για να επιτευχθεί η εξαγωγή του περιγράμματος (σχήματος) αντικειμένων που βρίσκονται σε μία εικόνα ή ένα καρέ ενός βίντεο. Στη συνέχεια τα σχήματα που εξάγονται δίνονται ως είσοδο σε ένα ήδη υπάρχον σύστημα [7] το οποίο επιστρέφει ένα ποσοστό ομοιότητας των σχημάτων σε σχέση με κάποιο από αυτά.

Η διεπαφή χρήστη επιτρέπει την εξαγωγή καρέ από το βίντεο είτε αυτόματα είτε κατ' επιλογή. Επίσης μετά την ολοκλήρωση της διαδικασίας ο χρήστης μπορεί να επανεμφανίσει τα καρέ που βρέθηκαν να έχουν το μεγαλύτερο ποσοστό ομοιότητας.

Στο *σχήμα 4.1* περιγράφεται η λειτουργία του συστήματος. Η εφαρμογή Video Shape Extractor δέχεται ως είσοδο μία εικόνα (ερώτηση) και ένα βίντεο. Από το βίντεο εξάγονται τα καρέ ως εικόνες με χρήση του Frame Grabbing Control (FGC) της βιβλιοθήκης JMF. Στην εικόνα – ερώτηση και στις εικόνες που προέκυψαν από το βίντεο, εφαρμόζονται οι αλγόριθμοι εξαγωγής σχήματος από όπου και εξάγεται

ένα περίγραμμα για κάθε εικόνα. Οι συντεταγμένες των περιγραμμάτων αποθηκεύονται σε ένα αρχείο XML και δίνονται ως είσοδος στο εργαλείο υπολογισμού της ομοιότητας. Το τελευταίο υπολογίζει την ομοιότητα των καρέ του βίντεο με την εικόνα – ερώτηση και επιστρέφει τα αποτελέσματα σε ένα αρχείο κειμένου. Το αρχείο αυτό περιέχει τα ποσοστά ομοιότητας των καρέ που έχουν όμοια σχηματική πληροφορία με την εικόνα – ερώτηση. Το εργαλείο VSX διαβάζει τα αποτελέσματα αυτά και με τη βοήθεια του Frame Positioning Control (FPC) εμφανίζει τα καρέ που παρουσίασαν ομοιότητα.



Σχήμα 4.1 – Περιγραφή της λειτουργίας του συστήματος

Στην παράγραφο 4.1 περιγράφεται η διαδικασία εξαγωγής του σχήματος από μία εικόνα ή από ένα καρέ ενός βίντεο Στην παράγραφο 4.2 παρουσιάζονται μερικές βασικές λειτουργίες του Java Media Framework που αξιοποιούνται από την εφαρμογή. Στην παράγραφο 4.3, παρουσιάζεται συνοπτικά το σύστημα ανάκτησης – CV_TOOL και περιγράφεται ο τρόπος διασύνδεσης του με το σύστημα εξαγωγής σχήματος και στην παράγραφο 4.4 παρουσιάζονται οι βασικές λειτουργίες της διεπαφής χρήστη.

4.1. Διαδικασία εξαγωγής σχήματος από εικόνα.

Από τη στιγμή που έχει τελειώσει η εξαγωγή των καρέ από ένα ή περισσότερα βίντεο, ακολουθεί η επεξεργασία του κάθε καρέ μεμονωμένα για την εξαγωγή του σχήματος. Για την επίτευξη του τελευταίου, εκτελούνται μία σειρά από λειτουργίες οι οποίες παρουσιάζονται παρακάτω.

4.1.1. Mean Filter (Φίλτρο Μέσου).

Το φίλτρο μέσου είναι ένα απλό φίλτρο που χρησιμοποιείται για την εξομάλυνση και την ελάττωση του θορύβου σε εικόνες. Το φίλτρο αντικαθιστά την τιμή του κάθε pixel της εικόνας με το μέσο όρο των γειτονικών pixels συμπεριλαμβανομένου και του ιδίου. Αυτό έχει σαν αποτέλεσμα την εξάλειψη των τιμών των pixel που δεν ταιριάζουν στην περιοχή τους. Το φίλτρο αυτό είναι ένα φίλτρο συνέλιξης. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ένας πίνακας 3x3 του οποίου όλες οι τιμές είναι 1/9. Στο παρακάτω σχήμα (Σχήμα 4.2) παρουσιάζεται ο πίνακας συνέλιξης.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Σχήμα 4.2 – Πίνακας Συνέλιξης Φίλτρου Μέσου

Το φίλτρο χρησιμοποιείται για εξομάλυνση και ελάττωση του θορύβου. Μετά την εφαρμογή του παρατηρείται στην εικόνα ένα θάμπωμα. Το φίλτρο κάνει αποτελεσματικότερη την τμηματοποίηση (segmentation) της εικόνας, που είναι το επόμενο βήμα. Στην εικόνα 4.1 παρουσιάζεται η εφαρμογή του φίλτρου σε μία εικόνα κατά τη διάρκεια της εκτέλεσης της εφαρμογής.



Εικόνα 4.1 – Αποτέλεσμα της εφαρμογής του Φίλτρου Μέσου
κατά τη διάρκεια εκτέλεσης της εφαρμογής.

4.1.2. K-means clustering

Ο αλγόριθμος k-means clustering [8] είναι ένας ανεπίβλεπτος αλγόριθμος εκμάθησης και χρησιμοποιείται για την αντιμετώπιση του προβλήματος των συστάδων (clustering). Η διαδικασία κατηγοριοποιεί ένα σύνολο δεδομένων σε έναν αριθμό από συστάδες. Η κεντρική ιδέα είναι να καθοριστούν K κέντρα, ένα για κάθε συστάδα. Έχει σημασία ο τρόπος με τον οποίο αυτά τα κέντρα θα τοποθετηθούν, αφού η τοποθέτησή τους σε διαφορετικές θέσεις προκαλεί και διαφορετικά αποτελέσματα. Η καλύτερη επιλογή θα ήταν να τοποθετηθούν όσο το δυνατόν πιο μακριά το ένα από το άλλο. Το επόμενο βήμα είναι να συνδεθεί το κάθε σημείο που ανήκει σε ένα σύνολο στοιχείων με το κοντινότερο κέντρο. Όταν συνδεθούν όλα τα σημεία, το πρώτο βήμα έχει ολοκληρωθεί. Σε αυτό το σημείο πρέπει να επανυπολογιστούν νέα κέντρα και αφού γίνει αυτό πραγματοποιείται νέα σύνδεση μεταξύ των σημείων του κάθε συνόλου δεδομένων με το κοντινότερο κέντρο. Τα K κέντρα αλλάζουν θέση βήμα προς βήμα έως ότου να μη χρειάζονται άλλες μετακινήσεις.

Η λειτουργία του αλγορίθμου προκύπτει από τον τύπο

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

Όπου το $\|x_i^{(j)} - c_j\|^2$ είναι ένα επιλεγμένο μέτρο απόστασης ανάμεσα σε ένα σημείο $x_i^{(j)}$ και το κέντρο c_j , είναι ένας δείκτης της απόστασης των n σημείων από τα αντίστοιχα κέντρα.

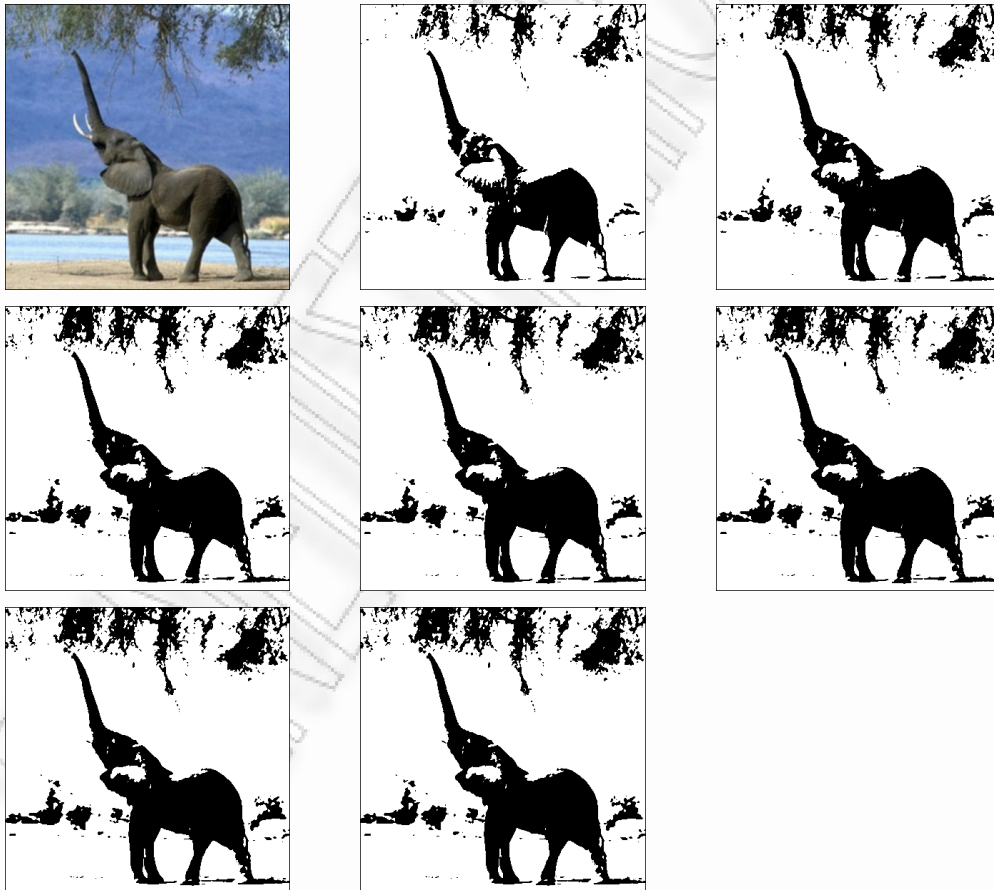
Ο αλγόριθμος χρησιμοποιείται σε αυτό το σημείο για να απομονωθούν τα αντικείμενα που βρίσκονται στην εικόνα από το φόντο. Η πληροφορία που προκύπτει μετά το τέλος της επεξεργασίας της εικόνας από τον αλγόριθμο, είναι μία εικόνα δύο χρωμάτων (άσπρο / μαύρο).

Στη συνέχεια περιγράφονται τα βήματα του αλγορίθμου.

1. Επιλέγονται δύο τυχαία pixels από την εικόνα, τα οποία αναπαριστούν τα αρχικά κέντρα των δύο ομάδων (cluster centers).

2. Τα pixels της εικόνας συγκρίνονται ένα προς ένα με τα δύο pixels – cluster centers και στη συνέχεια γίνεται αντιστοίχιση του κάθε pixel της εικόνας με το κέντρο από το οποίο έχει τη μικρότερη χρωματική απόσταση.
3. Υπολογίζεται η μέση τιμή από τα αντιστοιχισμένα pixels για την κάθε ομάδα και ορίζονται νέα κέντρα.
4. Υπολογίζεται το συνολικό άθροισμα των αποστάσεων μεταξύ των τιμών των pixels και των cluster centers για να υπολογιστεί αν τα cluster centers μετακινούνται. Όσο τα cluster centers μετακινούνται τα βήματα 2 και 3 επαναλαμβάνονται.

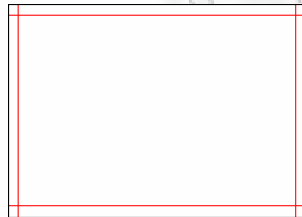
Στην εικόνα 4.2 παρουσιάζεται η εφαρμογή του αλγόριθμου κατά τη διάρκεια της εξέλιξης της ποριάς της εφαρμογής.



Εικόνα 4.2 – Αποτελέσματα της εκτέλεσης του αλγόριθμου K-Means κατά τη διάρκεια εκτέλεσης της εφαρμογής.

Το αποτέλεσμα του αλγόριθμου όπως έχει διαμορφωθεί για τη συγκεκριμένη εφαρμογή είναι μία δυαδική εικόνα (άσπρο – μαύρο). Η εικόνα αυτή μπορεί να είναι είτε με το φόντο μαύρο και τα αντικείμενα μέσα λευκά είτε το αντίστροφο. Το επιθυμητό αποτέλεσμα είναι λευκό φόντο με μαύρο αντικείμενο.

Για να λυθεί αυτό το πρόβλημα, υπολογίζονται κάθε φορά τα pixels των πλευρών της εικόνας όπως φαίνεται στο σχήμα 4.3. Τα pixels των οποίων το άθροισμα έχει τη μεγαλύτερη τιμή, θα είναι αυτά του φόντου και θα έχουν λευκό χρώμα. Επιλέχθηκε να υπολογίζονται τα σημεία που βρίσκονται πάνω στις κόκκινες γραμμές, δηλαδή λίγο πιο μέσα από το περίγραμμα της εικόνας έτσι ώστε να αποφευχθεί λάθος υπολογισμός στην περίπτωση που η εικόνα είναι καρέ από κακής ποιότητας βίντεο με μαύρες μπάρες στις άκρες.



Σχήμα 4.3

4.1.3 Erosion – Dilation

Σε αυτό το σημείο χρησιμοποιούνται οι δύο βασικές λειτουργίες μορφολογίας erosion και dilation. Η λειτουργία erosion εφαρμόζεται κυρίως σε δυαδικές εικόνες, αλλά υπάρχουν εκδόσεις που λειτουργούν και σε εικόνες της κλίμακας του γκρι. Η βασική επίδραση της λειτουργίας σε μια δυαδική εικόνα είναι να συρρικνώσει τα όρια των περιοχών των pixels των αντικειμένων που βρίσκονται στο πρώτο πλάνο. Κατά συνέπεια οι τομείς των pixels πρώτου πλάνου συρρικνώνονται στο μέγεθος, και οι τρύπες μέσα σε εκείνες τις περιοχές γίνονται μεγαλύτερες.

Ο μαθηματικός ορισμός της λειτουργία erosion για τις δυαδικές εικόνες είναι ο ακόλουθος:

- Έστω X το σύνολο των συντεταγμένων που αντιστοιχούν στη δυαδική εικόνα και K το σύνολο συντεταγμένων μίας μάσκας (kernel).
- Έστω ακόμη K_x η μετάφραση του K ώστε η προέλευσή του να είναι το X .
- Έτσι το erosion του X από το K είναι το σύνολο όλων των σημείων X όπου το K_x είναι υποσύνολο του X .

Στο σχήμα 4.4.1 παρουσιάζεται μία μάσκα 3×3 και στο σχήμα 4.4.2 οι συντεταγμένες των σημείων. Τα pixels πρώτου πλάνου αναπαρίστανται από 1 ενώ τα pixels του φόντου από 0.

1	1	1
1	1	1
1	1	1

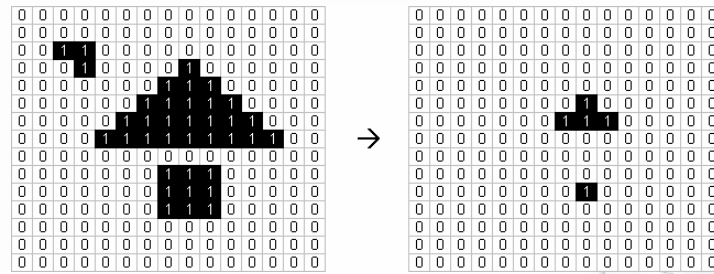
Σχήμα 4.4.1

(-1,-1)	(0,-1)	(1,-1)
(-1,0)	(0,0)	(1,0)
(-1,1)	(0,1)	(1,1)

Σχήμα 4.4.2

Για την εκτέλεση της λειτουργίας erosion σε μία δυαδική εικόνα με την παραπάνω μάσκα, εξετάζονται όλα τα pixels με τη σειρά. Για κάθε pixel που ανήκει στο πρώτο πλάνο της εικόνας, (pixel αντικειμένου), εφαρμόζεται η μάσκα πάνω στην εικόνα έτσι ώστε να συμπίπτει με τις συντεταγμένες του pixel του αντικειμένου. Εάν για κάθε pixel της μάσκας, το αντίστοιχο pixel στην εικόνα είναι pixel του αντικειμένου, τότε το pixel αντικειμένου αφήνεται όπως είναι. Εάν οποιαδήποτε από τα αντίστοιχα pixels στην εικόνα είναι pixels φόντου, τότε το pixel του αντικειμένου γίνεται pixel φόντου.

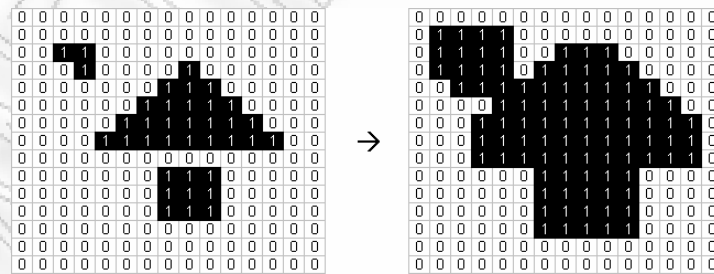
Η επίδραση αυτής της λειτουργίας είναι ουσιαστικά η αφαίρεση οποιουδήποτε pixel αντικειμένου δεν περιβάλλεται εντελώς από άλλα pixels. Τέτοια pixels βρίσκονται στις άκρες των μαύρων περιοχών, και έτσι το πρακτικό αποτέλεσμα είναι ότι οι περιοχές αυτές να συρρικνώνονται και οι τρύπες μέσα σε μια περιοχή να αυξάνονται. Στο σχήμα 4.5 παρουσιάζεται το αποτέλεσμα της λειτουργίας erosion με μάσκα 3×3 .



Σχήμα 4.5 - Αποτέλεσμα της λειτουργίας erosion με μάσκα 3x3.

Η λειτουργία dilation είναι η αντίθετη της erosion. Εφαρμόζεται με τον ίδιο ακριβώς τρόπο αλλά η επίδρασή της είναι η διαστολή των ορίων των περιοχών των pixels των αντικειμένων που βρίσκονται στο πρώτο πλάνο. Κατά συνέπεια οι τομείς των pixels πρώτου πλάνου διαστέλλονται στο μέγεθος, και οι τρύπες μέσα σε εκείνες τις περιοχές γίνονται μικρότερες. Η μάσκα εφαρμόζεται με τον ίδιο τρόπο πάνω στην εικόνα. Εάν για κάθε pixel της μάσκα, το αντίστοιχο pixel στην εικόνα είναι pixel του αντικειμένου, τότε το pixel αντικειμένου αφήνεται όπως είναι. Εάν οποιαδήποτε από τα αντίστοιχα pixels στην εικόνα είναι pixels φόντου, τότε το pixel του φόντου γίνεται pixel αντικειμένου.

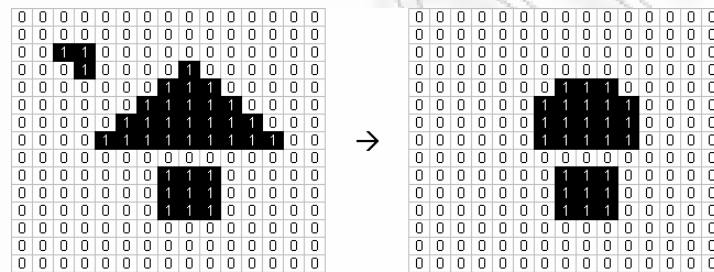
Η επίδραση αυτής της λειτουργίας είναι να θέσει στο χρώμα του αντικειμένου οποιαδήποτε pixels φόντου έχουν ένα γειτονικό pixel αντικειμένου. Το πρακτικό αποτέλεσμα είναι η αύξηση των περιοχών του αντικειμένου. Στο σχήμα 4.6 παρουσιάζεται το αποτέλεσμα της λειτουργίας dilation με μάσκα 3x3.



Σχήμα 4.6 - Αποτέλεσμα της λειτουργίας dilation με μάσκα 3x3.

Η σειρά με την οποία εκτελούνται οι δύο λειτουργίες είναι πρώτα η erosion και μετά η dilation έτσι ώστε να εκτελεστεί η λειτουργία opening. Ενώ η erosion μπορεί να χρησιμοποιηθεί για να αποβάλει ανεπιθύμητα pixels πρώτου πλάνου (π.χ. salt

noise), αρκετά αποτελεσματικά, έχει το μεγάλο μειονέκτημα ότι έχει επιπτώσεις σε όλες τις περιοχές των pixels πρώτου πλάνου αδιακρίτως. Η λειτουργία opening αντιμετωπίζει αυτό το πρόβλημα εκτελώντας τις δύο λειτουργίες τη μία μετά την άλλη. Όλα τα pixels που μπορούν να καλυφθούν από τη μάσκα με τη μάσκα να είναι εξ ολοκλήρου μέσα στην περιοχή πρώτου πλάνου θα συντηρηθούν. Όλα τα pixels πρώτου πλάνου που δεν μπορούν να καλυφθούν από τη μάσκα χωρίς μέρη της να βρίσκονται έξω από την περιοχή πρώτου πλάνου θα σβηστούν. Αφού πραγματοποιηθεί η λειτουργία, τα νέα όρια των περιοχών πρώτου πλάνου θα είναι τέτοια που η μάσκα θα ταιριάζει μέσα σε αυτά και οι περαιτέρω λειτουργίες opening με την ίδια μάσκα δεν θα έχουν καμία επίδραση. Η επίδραση ενός ανοίγματος σε μια δυαδική εικόνα που χρησιμοποιεί μία μάσκα 3x3 παρουσιάζεται στο Σχήμα 4.7.



Σχήμα 4.7 - Αποτέλεσμα της λειτουργίας opening με μάσκα 3x3.

Στην εικόνα 4.3 παρουσιάζεται η εφαρμογή του αλγόριθμου κατά τη διάρκεια της εξέλιξης της πορείας της εφαρμογής. Ο κύριος λόγος που χρησιμοποιήθηκε η λειτουργία αυτή σε αυτό το σημείο είναι η εξάλειψη των πολύ μικρών αντικείμεμων (θόρυβος).



Εικόνα 4.3 – Αποτελέσματα της εκτέλεσης της λειτουργίας opening κατά τη διάρκεια εκτέλεσης της εφαρμογής.

4.1.4. Object Isolation

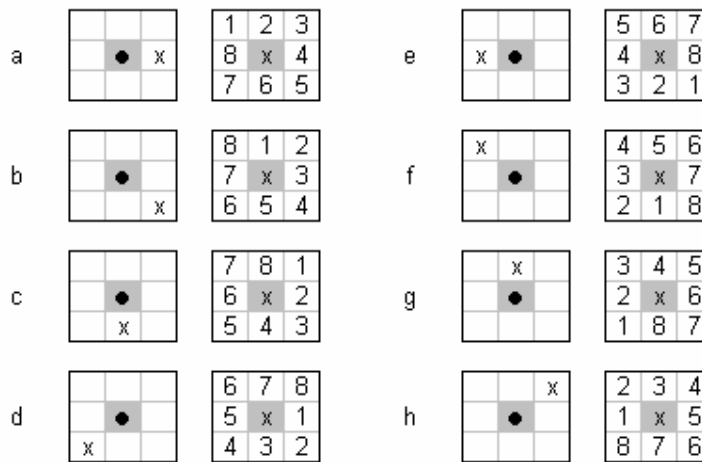
Σε αυτό το σημείο απομονώνεται από την εικόνα το μεγαλύτερο αντικείμενο και τα υπόλοιπα μικρότερα διαγράφονται. Ο αλγόριθμος εφαρμόζεται σε δυαδική εικόνα. Ακολουθούν τα βήματα του αλγόριθμου που υλοποιεί αυτή τη λειτουργία.

1. Η εικόνα σαρώνεται pixel προς pixel μέχρι να βρεθεί ένα μαύρο pixel. (δηλ. pixel που ανήκει σε ένα από τα αντικείμενα που βρίσκονται στην εικόνα).
2. Ελέγχονται τα γειτονικά pixels του αντικειμένου και στη συνέχεια τα γειτονικά των γειτονικών μέχρι να προσδιοριστεί το μέγεθος του αντικειμένου.
3. Γίνεται έλεγχος αν έχει βρεθεί προηγούμενο αντικείμενο με μεγαλύτερο μέγεθος από αυτό. Αν όχι το αντικείμενο αποθηκεύεται προσωρινά.
4. Η σάρωση συνεχίζεται από το σημείο που είχε σταματήσει και τα βήματα 2 και 3 επαναλαμβάνονται μέχρι να υπολογιστεί το μέγεθος όλων των αντικειμένων. Επιστρέφεται το τελευταίο αντικείμενο που έχει αποθηκευτεί.

Πιο αναλυτικά, όπως αναφέρθηκε και προηγουμένως, η εικόνα σαρώνεται μέχρι να βρεθεί ένα pixel που ανήκει σε αντικείμενο της εικόνας (βήμα 1). Από τη στιγμή που ένα τέτοιο pixel βρεθεί, ελέγχονται τα οκτώ γειτονικά του pixels για να αποφασιστεί αν αυτά ανήκουν στο αντικείμενο ή όχι με τη σειρά που φαίνεται στο σχήμα 4.8.1. Αν κάποια από τα pixels αυτά ανήκουν στο αντικείμενο τότε καταγράφονται για περαιτέρω έλεγχο. Αφού για ένα pixel, πραγματοποιηθεί ο έλεγχος των γειτονικών, αποθηκεύεται η θέση του σε μία δομή, αυξάνεται ο αριθμός των pixels του αντικειμένου κατά ένα και τέλος το σημείο διαγράφεται από την εικόνα ώστε να μην ελεγχθεί δεύτερη φορά (βήμα 2). Ο έλεγχος για κάθε αντικείμενο σταματάει μόλις ελεγχθούν όλα του τα pixels. Τότε γίνεται ένας ακόμα έλεγχος για να φανεί αν το αντικείμενο αυτό έχει περισσότερα pixels (δηλ. αν είναι μεγαλύτερο) από κάποιο προηγούμενο (βήμα 3). Η σάρωση συνεχίζεται από το σημείο που είχε σταματήσει και τα βήματα 2 και 3 επαναλαμβάνονται μέχρι να εξεταστούν όλα τα pixels της εικόνας.

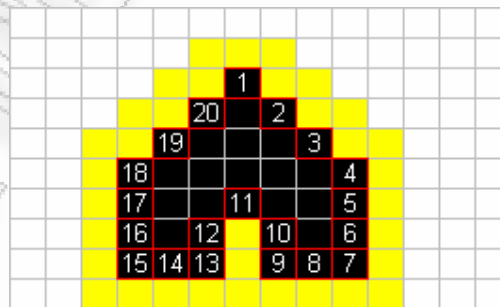
Στο σχήμα 4.8.2, παρουσιάζεται ένα παράδειγμα της λειτουργίας του αλγόριθμου για μία εικόνα 10 x 14 pixels. Με το μαύρο χρώμα είναι τα pixels των αντικειμένων και οι αριθμοί επάνω τους αποτελούν τη σειρά με την οποία πρόκειται να εξεταστούν.

του προηγούμενου σημείου. Βάσει αυτού, στη δεύτερη στήλη παρουσιάζεται με αριθμούς η σειρά των γειτονικών pixels που πρόκειται να εξεταστούν. Μόλις εντοπιστεί ένα νέο pixel του περιγράμματος της εικόνας, επαναλαμβάνεται η ίδια διαδικασία. Ο αλγόριθμος, μόλις εντοπιστεί το πρώτο pixel του περιγράμματος, εκτελεί την περίπτωση “a”. Καθ’ όλη τη διάρκεια της εκτέλεσης του αλγόριθμου, τα pixels που εντοπίζονται αποθηκεύονται με μία δομή και επιστρέφονται σε μορφή εικόνας κατά τον τερματισμό του.



Σχήμα 4.9

Στο σχήμα 4.10 παρουσιάζεται ένα παράδειγμα εκτέλεσης του αλγορίθμου για μία εικόνα 14 x 10 pixels. Τα αριθμημένα τετράγωνα με κόκκινο περίγραμμα είναι τα pixels του περιγράμματος. Τα pixels με κίτρινο χρώμα είναι αυτά που εξετάζονται για πιθανά σημεία του περιγράμματος χωρίς όμως να είναι.



Σχήμα 4.10 – Παράδειγμα λειτουργίας του αλγόριθμου Outline Extraction

Τέλος, στην εικόνα 4.5 παρουσιάζεται η εφαρμογή του αλγόριθμου κατά τη διάρκεια της εκτέλεσης της εφαρμογής.



Εικόνα 4.5 – Αποτελέσματα της εκτέλεσης του αλγόριθμου Outline Extraction κατά τη διάρκεια εκτέλεσης της εφαρμογής.

4.2 Χρήση του Java Media Framework

Το Java Media Framework είναι μια βιβλιοθήκη της Java που επιτρέπει την προσθήκη ήχου και βίντεο σε εφαρμογές και applets. Το πακέτο αυτό επεκτείνει την πλατφόρμα της Java και επιτρέπει την ανάπτυξη πολυμέσικων εφαρμογών

Το JMF χρησιμοποιήθηκε από το παρόν σύστημα για την αναπαραγωγή και την επεξεργασία των βίντεο. Επίσης χρησιμοποιήθηκε για την εξαγωγή των καρέ των βίντεο ως εικόνες, στις οποίες στη συνέχεια εφαρμόζονται οι αλγόριθμοι που προαναφέρθηκαν. Οι δύο βασικοί μηχανισμοί ελέγχου που χρησιμοποιήθηκαν είναι ο FrameGrabbingControl και ο FramePositioningControl.

Ο FrameGrabbingControl είναι ένας μηχανισμός ελέγχου που δίνει τη δυνατότητα εξαγωγής ενός ή περισσοτέρων καρέ κατά τη διάρκεια ροής ενός βίντεο. Ο FramePositioningControl επιτρέπει τον ακριβή προσδιορισμό της θέσης των καρέ μέσα σε ένα βίντεο.

4.3 Το σύστημα ανάκτησης (CV_TOOL)

Το εργαλείο ανάκτησης CV_TOOL χρησιμοποιεί μια πρωτότυπη μέθοδο ταιριάσματος σχημάτων, την Συνάρτηση Διαφοράς Γωνίας Στροφής (Turning Function Difference – TFD), η οποία βασίζεται στην κυρτότητα του περιγράμματος του σχήματος και είναι αμετάβλητη ως προς την μετάθεση, την περιστροφή και το μέγεθος. Η μέθοδος παράγει πληροφορίες που αφορούν την αντιστοιχία των σημείων

που ανήκουν στα σχήματα που συγκρίνονται κατά τη διάρκεια της διαδικασίας ερμηνείας. Η μέθοδος TFD εξηγεί τα αποτελέσματά της μέσω μιας ευθυγράμμισης και μιας οπτικής διαδικασίας animation που δίνει έμφαση στις ομοιότητες μεταξύ των πρότυπων εικόνων και κάθε μίας εκ των επιλεγμένων εικόνων που επεξεργάζονται από τη μέθοδο.

Το CV_TOOL χρησιμοποιεί μια διαδικασία ανάκτησης σχήματος πολλαπλών βημάτων η οποία εφαρμόζει επαναληπτικά κριτήρια ελέγχου στην αρχική βάση δεδομένων, μέχρι να παραμείνουν μόνο τα στοιχεία που περνούν όλες τις δοκιμές. Στη συνέχεια τα στοιχεία αυτά ταξινομούνται βάσει της ομοιότητας και είναι έτοιμα για την παρουσίαση. Η διαδικασία ταιριάσματος δέχεται ως είσοδο δύο πίνακες τιμών TFD που αναπαριστούν τα περιγράμματα των σχημάτων που συγκρίνονται. Η μέθοδος δέχεται επίσης ένα σύνολο παραμέτρων. Η έξοδος της μεθόδου αποτελείται από την αντιστοιχία μεταξύ των περιοχών των πολυγώνων που θεωρούνται παρόμοιες και το βαθμό ταιριάσματος.

Η διαδικασία εύρεσης όμοιων περιοχών επιδιώκει να βρει τις αντίστοιχες περιοχές ομοιότητας μεταξύ δύο σχημάτων. Για το συγκεκριμένο σκοπό, το βήμα αυτό δέχεται ως είσοδο μια τιμή που δείχνει τη μέγιστη διαφορά μεταξύ των αντίστοιχων τιμών της συνάρτησης γωνίας στα δύο σχήματα με σκοπό αυτά να ταιριάζουν επιτυχώς. Επιπλέον, δέχεται ως είσοδο τον ελάχιστο αριθμό διαδοχικών τιμών της συνάρτησης γωνίας στροφής που μπορεί να είναι χαρακτηριστικό των αντιστοιχισμένων περιοχών των καμπυλών. Μια διαδικασία ακολουθίας-ελέγχου αρχίζει για κάθε ζευγάρι στοιχείων και ελέγχει αυτά και τα στοιχεία που τα ακολουθούν προκειμένου να φανεί εάν η απόσταση μεταξύ των αντίστοιχων στοιχείων των δύο σειρών είναι κάτω από την τιμή της μέγιστης διαφοράς.

Δύο ακολουθίες θεωρούνται συμβατές εάν περνούν δύο δοκιμές που είναι βασισμένες στην απόσταση και την περιστροφή. Η πρώτη δοκιμή συμβατότητας είναι αυτή της απόστασης. Για την διαδικασία ταιριάσματος, δύο ταιριαστές ακολουθίες πρέπει να τοποθετηθούν σε λογικές αποστάσεις από την άποψη των κορυφών. Δύο περιοχές σε ένα πολύγωνο που βρίσκονται κοντά δεν θα μπορούσαν να αντιστοιχηθούν με δύο άλλες περιοχές στο άλλο πολύγωνο που είναι αρκετά μακριά. Η δεύτερη δοκιμή συμβατότητας είναι η δοκιμή περιστροφής. Δύο περιοχές σε ένα πολύγωνο που είναι ομοίως προσανατολισμένες δεν θα μπορούσαν να αντιστοιχηθούν με δύο άλλες περιοχές στο άλλο πολύγωνο που δεν είναι. Για τον καθορισμό παρόμοιου προσανατολισμού πρέπει αρχικά να καθοριστεί η αλλαγή του

προσανατολισμού μεταξύ δύο κορυφών. Η αλλαγή του προσανατολισμού μεταξύ κορυφών X, Y (του ίδιου πολυγώνου), ισούται με το άθροισμα των τιμών TFD που συναντούνται σε ένα δεξιόστροφο πολύγωνο διατρέχοντας από το X στο Y.

Μετά το τελικό βήμα ταιριάσματος της ακολουθίας, το αποτέλεσμα ομοιότητας μπορεί να υπολογιστεί με το άθροισμα των μηκών των ταιριαστών ακολουθιών και έπειτα τη διαίρεση με το ελάχιστο μήκος των δύο αντιστοιχισμένων πολυγώνων (σε περίπτωση που τα μήκη δεν είναι ίσα). Είναι προφανές ότι εάν όλα τα πολύγωνα στη βάση δεδομένων επιλέγονται στο ίδιο μέγεθος, ο υπολογισμός του ταιριάσματος των αποτελεσμάτων θα είναι πιο αξιόπιστος

Διασύνδεση του συστήματος VSX με το σύστημα CV_TOOL

Το CV_TOOL δέχεται ως είσοδο ένα αρχείο XML με τις συντεταγμένες των περιγραμμάτων ενός συνόλου από εικόνες και επιστρέφει ως έξοδο ένα δεύτερο αρχείο με τα ποσοστά ομοιότητας για κάθε περίγραμμα.

Στη λίστα 4.1 παρουσιάζεται ένα παράδειγμα ενός αρχείου εισόδου.

Λίστα 4.1.

```
<?xml version="1.0" standalone="no"?>
  <PDB sampleSize="-1">
    <polygon points='131.0 36.0, 132.0 36.0, 133.0 36.0, ..., 134.0 36.0' />
    <polygon points='168.0 137.0, 168.0 138.0, 168.0 139.0, ..., 167.0 140.0' />
    <polygon points='280.0 162.0, 281.0 162.0, 282.0 163.0, ..., 234.0 36.0' />
    <polygon points='67.0 84.0, 68.0 83.0, 68.0 82.0, ..., 43.0 12.0' />
    ...
    ...
    ...

    <polygon points='137.0 170.0, 136.0 169.0, 135.0 168.0, ..., 323.0 236.0' />
    <polygon points='177.0 41.0, 178.0 41.0, 179.0 41.0, ..., 321.0 123.0' />
  </PDB>
```

Η πρώτη γραμμή από σημεία, είναι οι συντεταγμένες του περιγράμματος που χρησιμοποιείται ως ερώτημα για την αναζήτηση. Οι υπόλοιπες γραμμές είναι τα περιγράμματα που πρόκειται να συγκριθούν με το πρώτο για να προκύψει το ποσοστό ομοιότητας. Το σύστημα υπολογίζει την ομοιότητα των πολυγώνων και επιστρέφει τα αποτελέσματα σε ένα δεύτερο αρχείο όπως παρουσιάζεται στη λίστα 4.2.

Λίστα 4.2.

```
0 1
6 0.861443
7 0.759885
5 0.743228
```

Η πρώτη γραμμή της λίστας 4.2 αφορά το ίδιο το σχήμα – ερώτημα και δεν λαμβάνεται υπόψη. Τα υπόλοιπα τρία μας πληροφορούν πως τα σχήματα 6, 7 και 5 ταιριάζουν με το σχήμα –ερώτημα σε ποσοστό 86%, 75% και 74% αντίστοιχα.

Το σύστημα εξαγωγής σχήματος (VSX) για να επικοινωνήσει με το σύστημα ανάκτησης (CV_TOOL), δημιουργεί ένα αρχείο με τις συντεταγμένες των περιγραμμάτων που έχει εξάγει από τα βίντεο και στη συνέχεια καλεί το σύστημα ανάκτησης για να εκτελέσει τον έλεγχο. Όταν ο έλεγχος ολοκληρωθεί διαβάζει τα αποτελέσματα από το δεύτερο αρχείο του συστήματος ανάκτησης και εμφανίζει τα αντίστοιχα καρτέ όπως θα δούμε και στην επόμενη ενότητα.

4.4. Περιγραφή διεπαφής χρήστη

Η διεπαφή χρήστη υλοποιήθηκε με το Java Swing. Σκοπός της είναι να παρέχει έναν εύκολο τρόπο διαχείρισης εικόνων και βίντεο, εξαγωγή και αποθήκευση της σχηματικής πληροφορίας ώστε να γίνει είσοδος στο σύστημα ανάκτησης και τέλος αποτελεσματική παρουσίαση των αποτελεσμάτων.

Πιο αναλυτικά, η διεπαφή χρήστη:

- Δίνει τη δυνατότητα στο χρήστη να φορτώνει εικόνες και βίντεο με χρήση menu.
- Παρέχει τη δυνατότητα εξαγωγής του περιγράμματος από εικόνα και από τα καρτέ ενός βίντεο. Στη δεύτερη περίπτωση ο χρήστης μπορεί να επιλέξει είτε αυτόματη εξαγωγή των περιγραμμάτων των καρτέ των βίντεο είτε χειροκίνητη.
- Δίνει την επιλογή της εντολής υπολογισμού στο σύστημα ανάκτησης.
- Διαγράφει τα ήδη υποθηκευμένα σχήματα για να πραγματοποιηθούν νέοι έλεγχοι.
- Εμφανίζει λίστα με τα αποτελέσματα του συστήματος ανάκτησης. Κάνοντας click στα αποτελέσματα της λίστας εμφανίζεται το συγκεκριμένο καρτέ του βίντεο που συγκρίθηκε με την εικόνα ερώτηση.

5

Παρουσίαση της εφαρμογής.

Στο κεφάλαιο αυτό παρουσιάζεται η διεπαφή χρήστη. Γίνεται αναλυτική παρουσίαση των λειτουργιών της και μερικά παραδείγματα χρήσης της. Επίσης παρουσιάζονται μερικά ενδεικτικά πειραματικά αποτελέσματα.

5.1 Απαιτήσεις συστήματος.

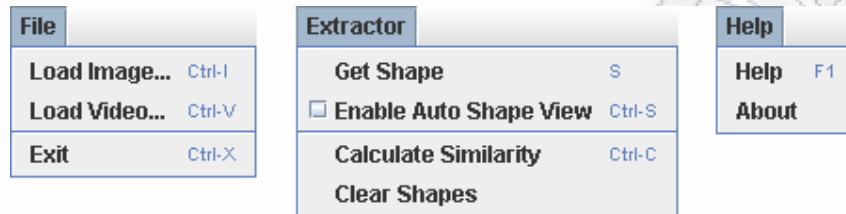
Για τη σωστή λειτουργία και εκτέλεση της εφαρμογής πρέπει να υπάρχει στο σύστημα εγκατεστημένο το περιβάλλον εκτέλεσης Java (έκδοση JRE-1.6.0, ή νεότερο). Επίσης πρέπει να υπάρχει εγκατεστημένη βιβλιοθήκη Java Media Framework (έκδοση JMF-2.1.1e).

Οι τύποι (format) των εικόνων που υποστηρίζονται είναι αυτοί που περιγράφονται στο JSR-015 Image I/O Framework Specification. Για την ορθή λειτουργία της εφαρμογής προτείνεται τα βίντεο να είναι σε ανάλυση 320x240 σε μορφή avi (απαιτείται να είναι εγκατεστημένα στο σύστημα τα απαραίτητα codecs).

Η εφαρμογή δοκιμάστηκε σε υπολογιστές Intel P4 1.6GHz και AMD 3400+ με περιβάλλον windows.

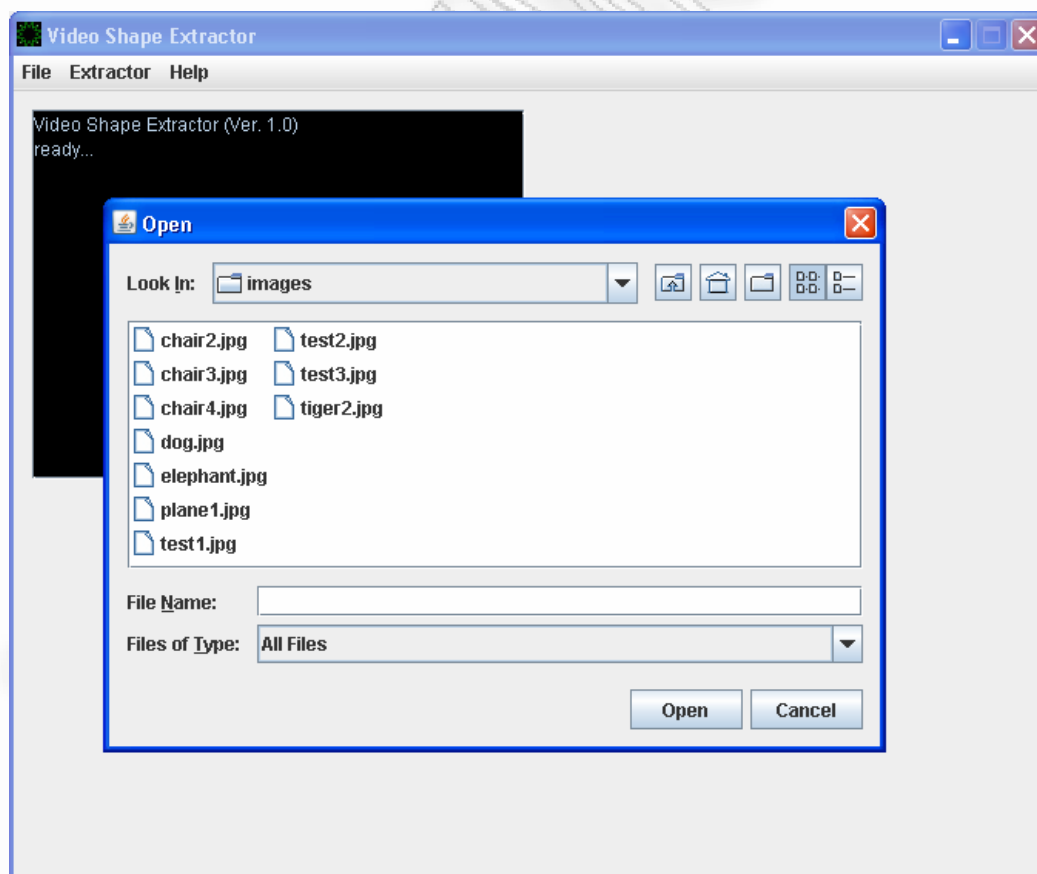
5.2 Παρουσίαση διεπαφής χρήστη (GUI)

Η γραμμή μενού χρησιμοποιείται από το χρήστη έτσι ώστε να μπορεί ο τελευταίος να δίνει εντολές στο σύστημα. Στο σχήμα 5.1 παρουσιάζεται το μενού της εφαρμογής.



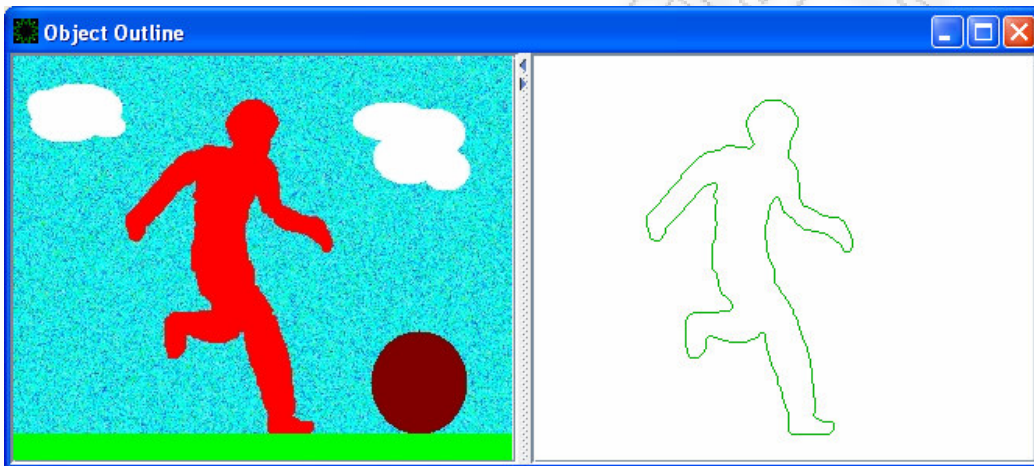
Σχήμα 5.1 – Το μενού της εφαρμογής.

Η εντολή “**Load Image**” του μενού “File” ανοίγει ένα πλαίσιο διαλόγου όπως φαίνεται στο σχήμα 5.2 και επιτρέπει στο χρήστη να διαλέξει μία εικόνα.



Σχήμα 5.2 – Άνοιγμα αρχείου εικόνας.

Το άνοιγμα μίας εικόνας είναι η πρώτη εντολή που πρέπει να εκτελεστεί. Με την επιλογή μίας εικόνας και το πάτημα του κουμπιού “Open” εκτελούνται μία σειρά από λειτουργίες. Φορτώνεται η εικόνα στη μνήμη και εκτελούνται με τη σειρά που παρουσιάστηκαν οι αλγόριθμοι που χρησιμοποιούνται για την εξαγωγή του σχήματος. Η εφαρμογή εμφανίζει ένα παράθυρο με την αρχική εικόνα και το περίγραμμα του αντικειμένου που “κρύβει” μέσα της (σχήμα 5.3). Επίσης το περίγραμμα που προέκυψε αποθηκεύεται για να χρησιμοποιηθεί ως σχήμα – ερώτηση για την ανάκτηση των σχημάτων που βρίσκονται μέσα στο βίντεο.

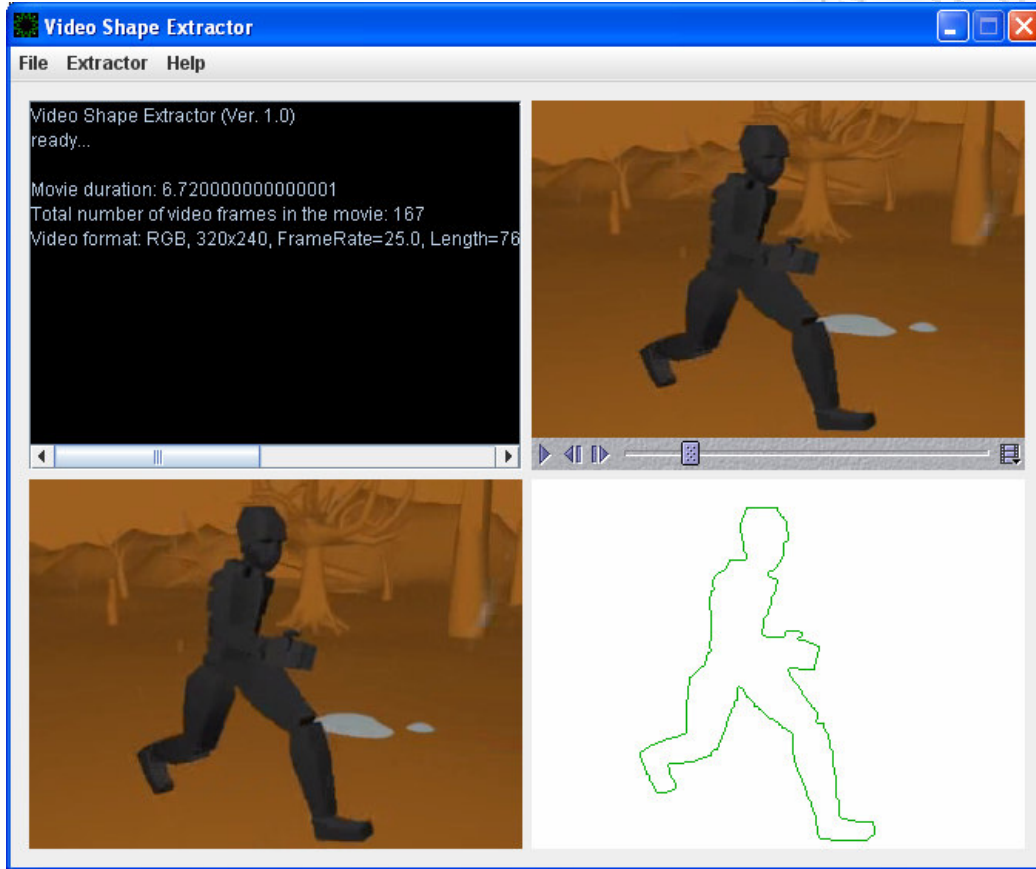


Σχήμα 5.3 – Άνοιγμα αρχείου εικόνας.

Το επόμενο βήμα είναι η εντολή “**Load Video**” η οποία φορτώνει ένα βίντεο για επεξεργασία. Με την εκτέλεση της εντολής ξεκινάει και η αναπαραγωγή του βίντεο. Από αυτή τη στιγμή ενεργοποιείται και το μενού “Extractor” και γίνονται διαθέσιμες οι εντολές που αφορούν την επεξεργασία του βίντεο.

Η εντολή “**Get Shape**” απομονώνει ένα καρέ του βίντεο και εφαρμόζει τους αλγόριθμους που χρησιμοποιούνται για την εξαγωγή του σχήματος. Η “**Enable Auto Shape View**” εκτελεί τις ίδιες ενέργειες όπως η “Get Shape” αλλά αυτόματα ανά δευτερόλεπτο (δηλαδή εξάγει εξήντα περιγράμματα ανά λεπτό). Τα περιγράμματα αποθηκεύονται για να δοθούν ως είσοδο στο σύστημα που πραγματοποιεί το έλεγχο της ομοιότητας. Στο *σχήμα 5.4* παρουσιάζεται ένα παράδειγμα της εντολής “Get Shape”. Στο πάνω αριστερά πλαίσιο του παραθύρου εμφανίζονται πληροφορίες σχετικές με το βίντεο. Στο πλαίσιο πάνω δεξιά πραγματοποιείται η αναπαραγωγή του

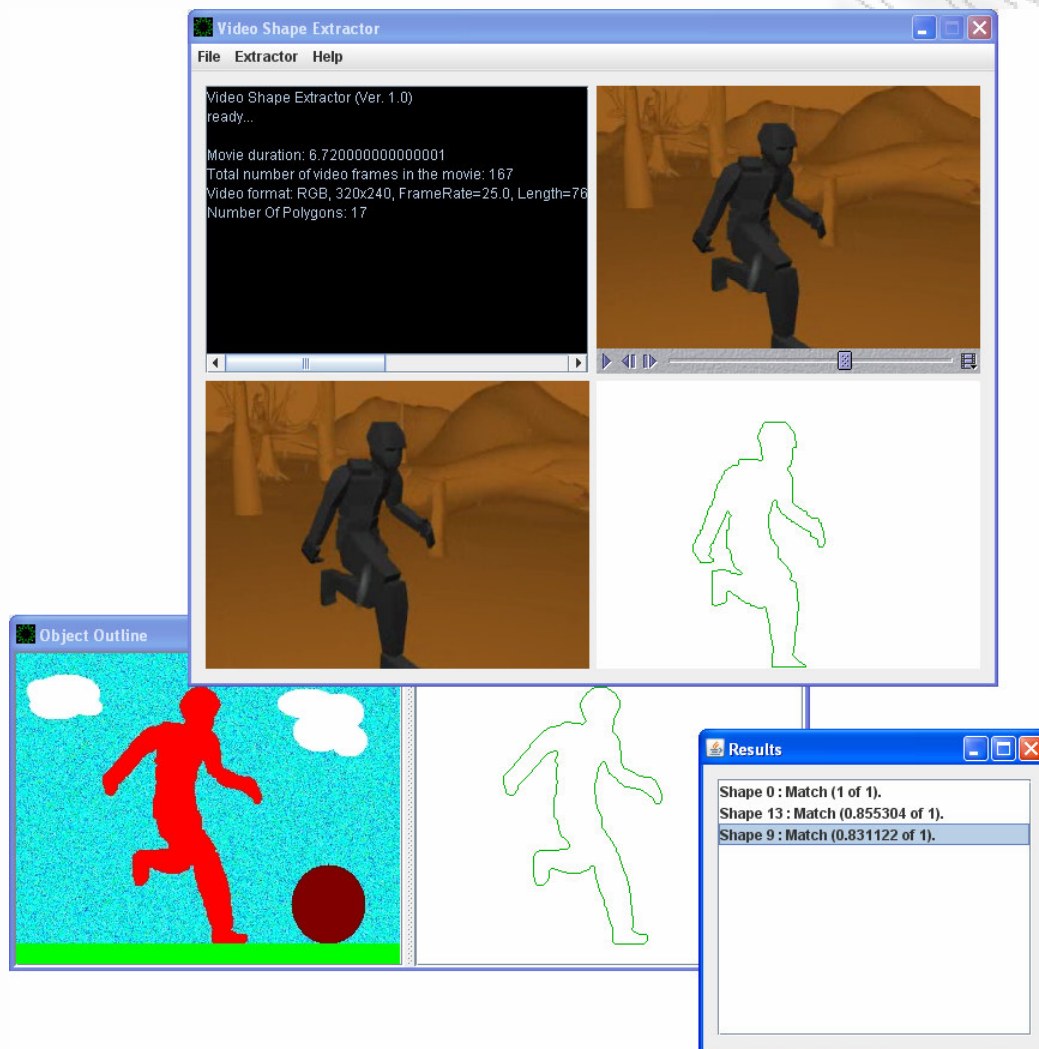
βίντεο. Στο πλαίσιο κάτω αριστερά εμφανίζεται το καρέ από το οποίο θα προκύψει το περίγραμμα του σχήματος και τέλος στο πλαίσιο κάτω δεξιά εμφανίζεται το περίγραμμα του σχήματος.



Σχήμα 5.4 – Παράδειγμα εξαγωγής σχήματος από καρέ ενός βίντεο.

Η εντολή “**Calculate Similarity**” δημιουργεί ένα αρχείο XML (*Λίστα 4.1*) με τις συντεταγμένες των σημείων από τα οποία αποτελούνται τα περιγράμματα που εξάγονται από την εικόνα – ερώτηση και τα καρέ του βίντεο. Στη συνέχεια καλείται το εργαλείο ανάκτησης και επιστρέφει τα αποτελέσματα σε μία λίστα όπως φαίνεται στο *σχήμα 5.5*. Κάνοντας click πάνω σε ένα από τα αποτελέσματα που εμφανίζονται στη λίστα, στο πλαίσιο αναπαραγωγής του βίντεο εμφανίζεται το συγκεκριμένο καρέ από το οποίο βγήκε το περίγραμμα. Στο πλαίσιο του περιγράμματος εμφανίζεται και το περίγραμμα.

Η εντολή “**Clear Shapes**” αδειάζει τη μνήμη από τα περιγράμματα που έχουν εξαχθεί μέχρι στιγμής με σκοπό να πραγματοποιηθεί νέο πείραμα.

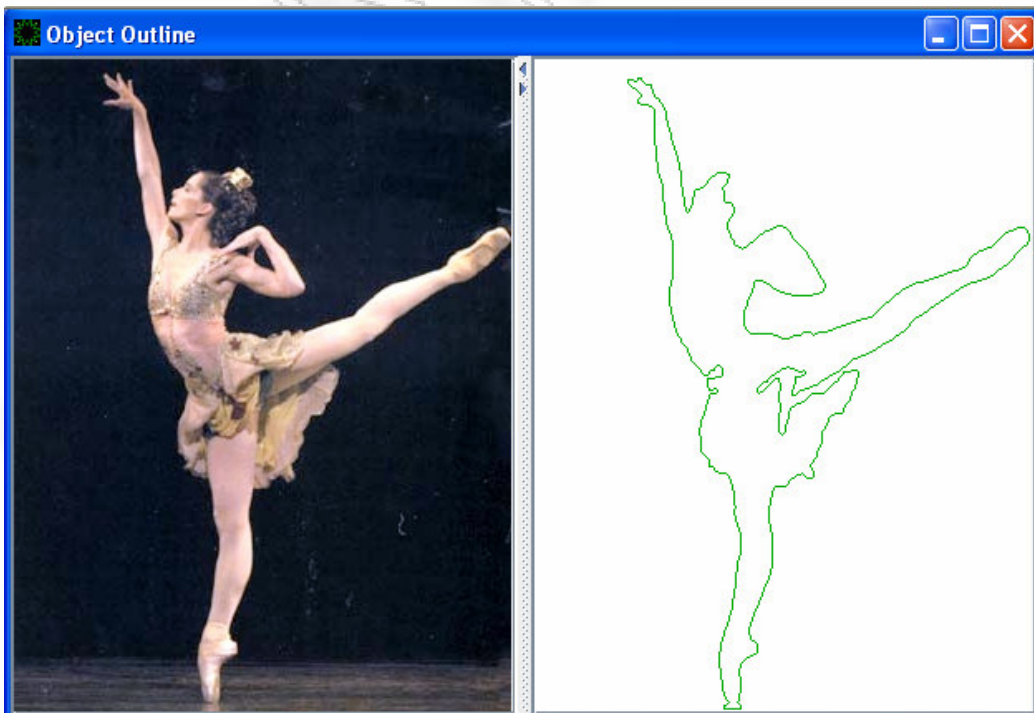
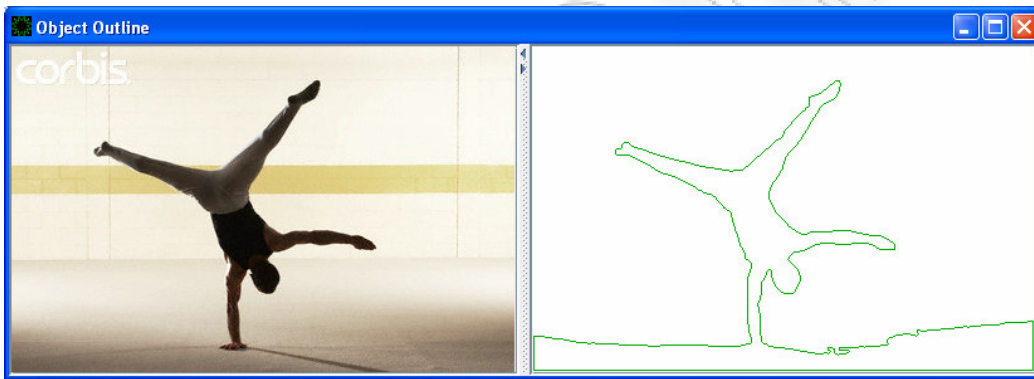
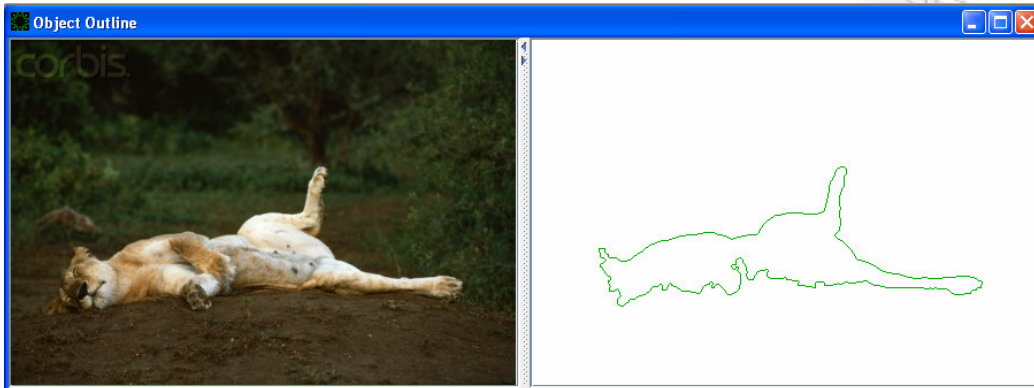


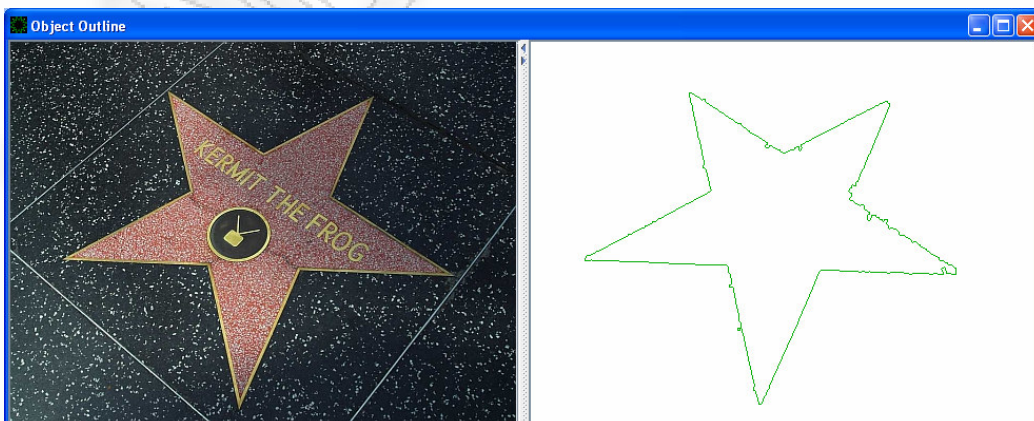
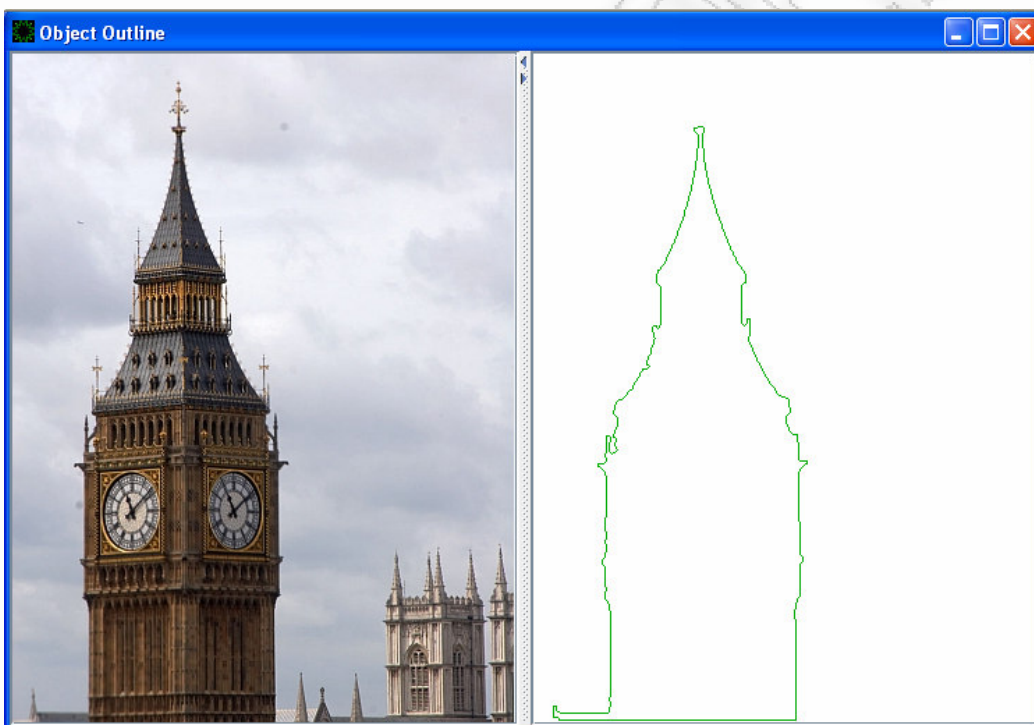
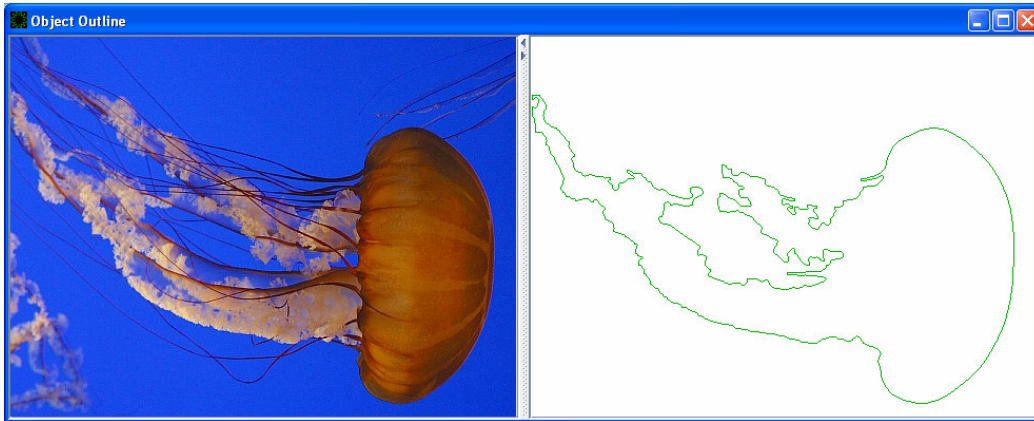
Σχήμα 5.5 – Παράδειγμα λειτουργίας της εφαρμογής.

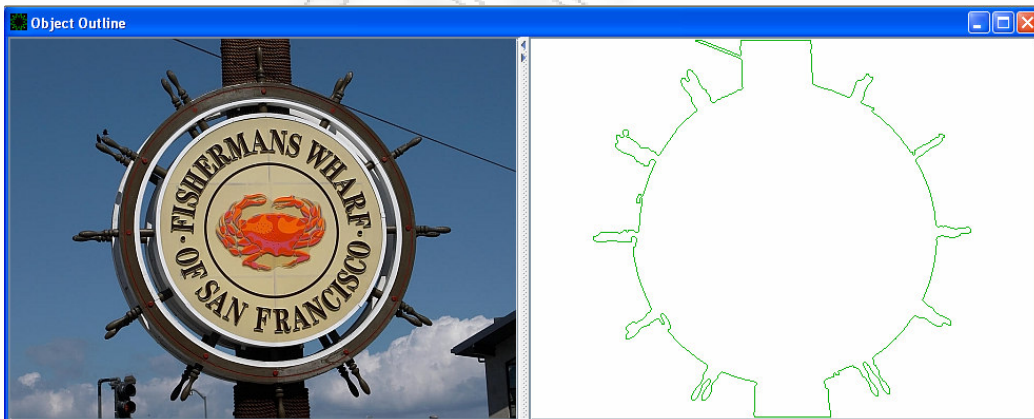
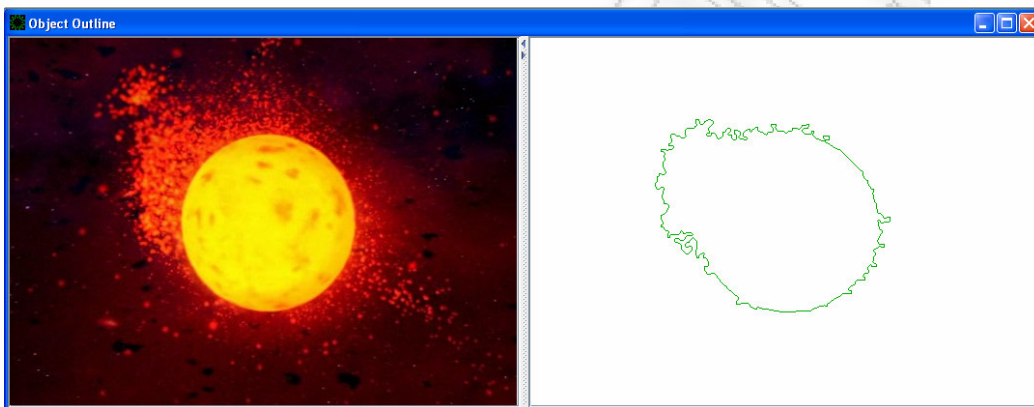
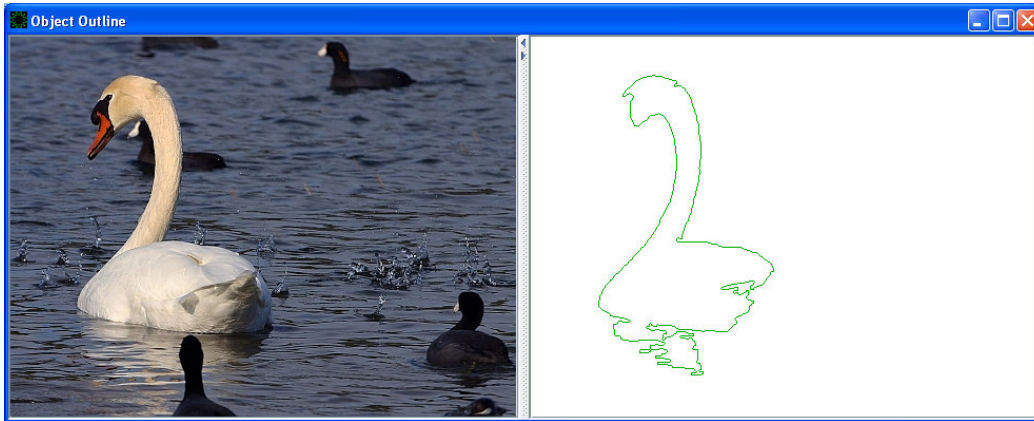
5.3 Πειραματικά Αποτελέσματα

Στην παράγραφο αυτή παρουσιάζονται μερικά πειραματικά αποτελέσματα εξαγωγής σχήματος από εικόνες και βίντεο με χρήση του συστήματος που αναπτύχθηκε. Επίσης στην παράγραφο 5.3.3 παρουσιάζονται παραδείγματα ανάκτησης με βάση τη σχηματική πληροφορία που εξήχθη.

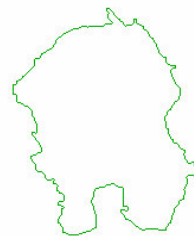
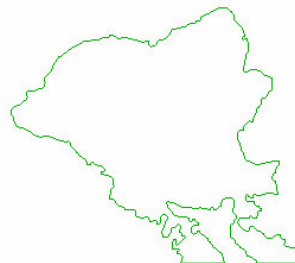
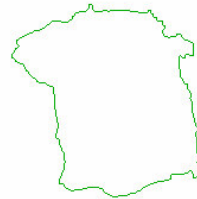
5.3.1 Αποτελέσματα εξαγωγής σχήματος από εικόνες

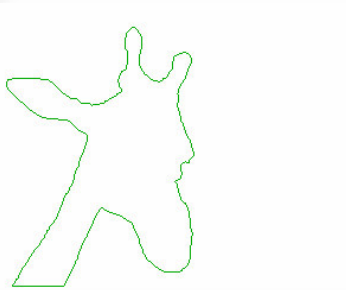
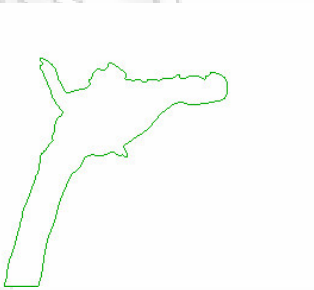
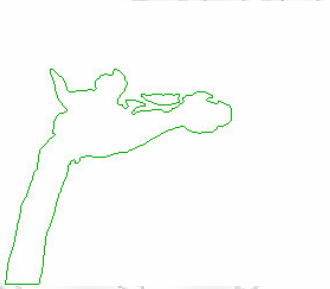
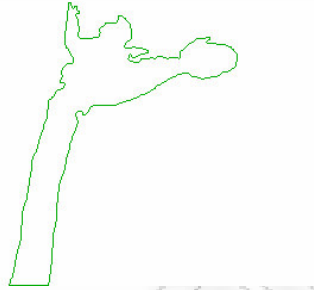






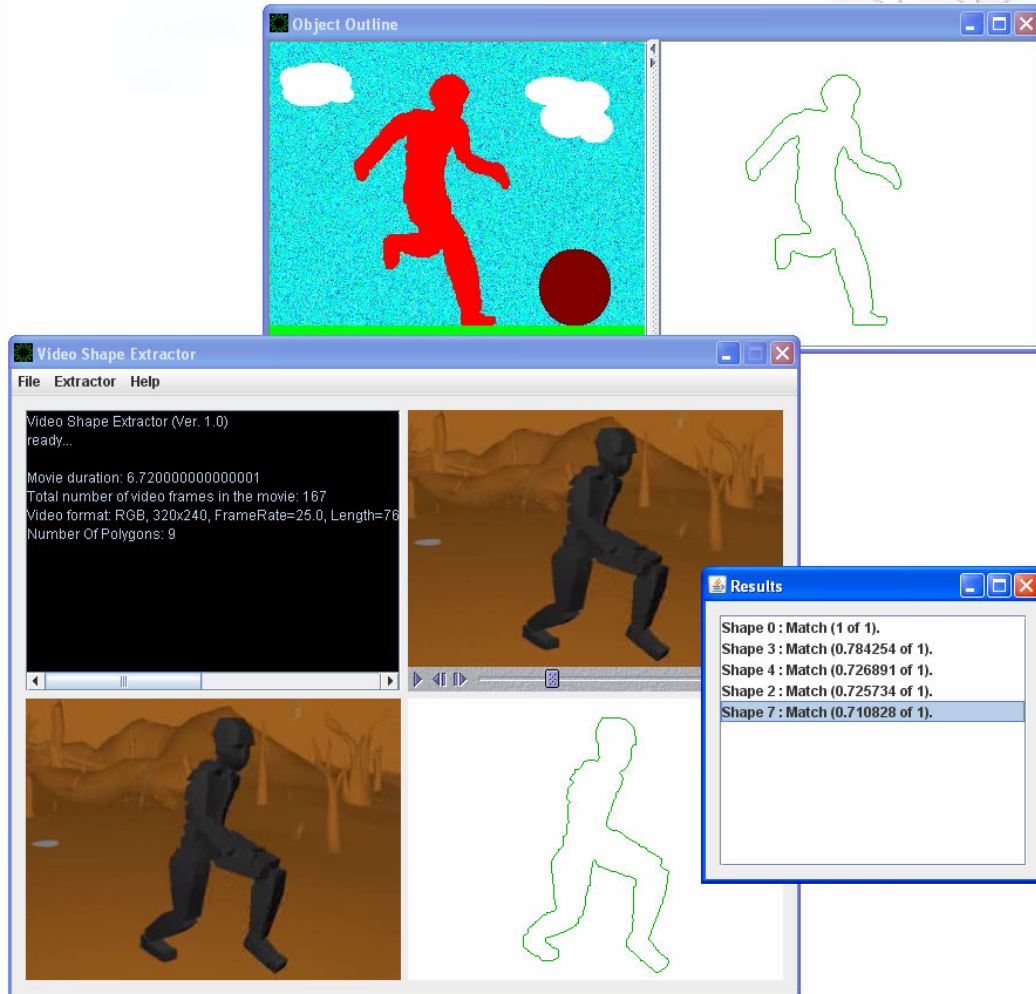
5.3.2 Αποτελέσματα εξαγωγής σχήματος από βίντεο





5.3.3 Αποτελέσματα ανάκτησης βίντεο

Παράδειγμα 1



Στιγμιότυπο κατά την εκτέλεση της εφαρμογής.



Shape1 – Match 78%.



Shape1 – Match 73%.

Παράδειγμα 2

Object Outline

Video Shape Extractor

File Extractor Help

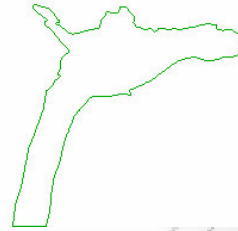
Video Shape Extractor (Ver. 1.0)
ready...

Movie duration: 14.440000000000001
Total number of video frames in the movie: 360
Video format: RGB, 320x240, FrameRate=25.0, Length=76
Number Of Polygons: 7

Results

Shape 0 : Match (1 of 1).
Shape 1 : Match (0.851761 of 1).
Shape 6 : Match (0.758466 of 1).

Στιγμιότυπο κατά την εκτέλεση της εφαρμογής.



Shape1 – Match 85%.



Shape2 – Match 76%.

PAPERKIT

6

Συμπεράσματα – προοπτικές

Ο κύριος στόχος της εργασίας αυτής ήταν η αναζήτηση ενός αποτελεσματικού τρόπου εξαγωγής σχηματικής πληροφορίας από βίντεο, καθώς και η μελέτη των υπαρχόντων συστημάτων ανάκτησης βίντεο.

Μελετώντας σε πρώτη φάση τις πιθανές τεχνικές εξαγωγής σχήματος από βίντεο, εύκολα μπορεί να συμπεράνει κανείς πως βρίσκεται αντιμέτωπος με ένα αντικείμενο το οποίο μελετάται εδώ και αρκετά χρόνια και το οποίο δεν έχει ακόμα λυθεί αποτελεσματικά για κάθε δυνατή περίπτωση. Επίσης παρατηρώντας τις μεμονωμένες περιπτώσεις προκύπτει ένα ακόμα συμπέρασμα, πως η εξαγωγή του σχήματος και των χαρακτηριστικών γνωρισμάτων από μία εικόνα ή ένα βίντεο είναι μία μεγάλη πρόκληση καθώς τα προβλήματα που αναζητούν μία σχετική λύση είναι απεριόριστα. Ενδεικτικά μπορεί να αναφέρει κανείς εφαρμογές στην ιατρική, στη βιομηχανία, σε συστήματα GIS αλλά ακόμα και στις καθημερινές υπηρεσίες που χρησιμοποιούν χρήστες του Διαδικτύου.

Στη συνέχεια αναζητώντας και αναλύοντας μερικά από τα υπάρχοντα συστήματα ανάκτησης εικόνας και βίντεο αποδεικνύεται και πάλι η ανάγκη για τη δημιουργία συστημάτων που θα μπορούν να καλύψουν ένα ευρύτερο φάσμα προβλημάτων ανάκτησης.

Το σύστημα που αναπτύχθηκε στα πλαίσια της εργασίας αυτής, καλύπτει ένα μέρος του προβλήματος της ανάκτησης βίντεο ικανοποιητικά. Παρόλα αυτά αρκετές είναι οι βελτιώσεις – επεκτάσεις που θα μπορούσε κανείς να ενσωματώσει στο σύστημα αυτό και αφορούν δύο μέρη. Όσον αφορά το πρώτο μέρος, όπως έχει αναφερθεί προηγουμένως, η κύρια τεχνική που χρησιμοποιήθηκε για την εξαγωγή του σχήματος βασίζεται στη χρωματική πληροφορία. Θα μπορούσαν ενδεχομένως να χρησιμοποιηθούν και άλλες τεχνικές σε συνδυασμό με αυτή έτσι ώστε να προκύψουν καλύτερα αποτελέσματα. Το δύσκολο κομμάτι σε αυτό το σημείο είναι οι υψηλές απαιτήσεις σε υπολογιστική ισχύ (λαμβάνοντας υπόψη πως ένα μεγάλο βίντεο υψηλής ανάλυσης είναι ένας μεγάλος όγκος πληροφορίας).

Μία δεύτερη επέκταση που θα μπορούσε να γίνει σε αυτό το σύστημα είναι να αξιοποιηθεί και ο ήχος και η κίνηση, που είναι δύο βασικά χαρακτηριστικά γνωρίσματα του βίντεο και μπορούν να προσφέρουν πολύτιμη πληροφορία.

ΠΑΡΑΡΤΗΜΑ

A. Java Documentation

makourou.vsx.algorithms

Class Convolution

java.lang.Object

└─ **makourou.vsx.algorithms.Convolution**

```
public class Convolution
extends java.lang.Object
```

This class performs the convolution operation. The image is convolved with a standard kernel in order to perform the mean filter.

Since:

1.0

Constructor Summary

[Convolution](#)(java.awt.image.BufferedImage input)

The constructor of the class.

Method Summary

java.awt.image.BufferedImage	getOutput () Returns the output image.
void	process () This method performs image convolution with a standard kernel in order to apply the mean filter on source image.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Convolution

```
public Convolution(java.awt.image.BufferedImage input)
```

The constructor of the class. It is used to set the input image.

Parameters:

input - the input image

Method Detail

process

```
public void process()
```

This method performs image convolution with a standard kernel in order to apply the mean filter on source image.

getOutput

```
public java.awt.image.BufferedImage getOutput()
```

Returns the output image.

Returns:

the image after the convolution.

makourou.vsx.tools

Class ImageComp

```
java.lang.Object
```

```
└ java.awt.Component
```

```
└ makourou.vsx.tools.ImageComp
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable

```
public class ImageComp  
extends java.awt.Component
```

An object to store an image component.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class java.awt.Component

```
java.awt.Component.AccessibleAWTComponent,  
java.awt.Component.BaselineResizeBehavior,  
java.awt.Component.BltBufferStrategy,  
java.awt.Component.FlipBufferStrategy
```

Field Summary

static long	serialVersionUID
-------------	----------------------------------

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

ImageComp (java.awt.image.BufferedImage img)
Creates new instance of the class

Method Summary

java.awt.Dimension	getPreferredSize ()
void	paint (java.awt.Graphics g)

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds,

Field Detail

serialVersionUID

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

ImageComp

```
public ImageComp(java.awt.image.BufferedImage img)
```

Creates new instance of the class

Parameters:

img - A buffered image that will be painted into the component.

Method Detail

getPreferredSize

```
public java.awt.Dimension getPreferredSize()
```

Overrides:

getPreferredSize in class java.awt.Component

paint

```
public void paint(java.awt.Graphics g)
```

Overrides:

paint in class java.awt.Component

makourou.vsx.tools

Class ImageSplitPane

```
java.lang.Object
```

```
└ java.awt.Component
```

```
└ java.awt.Container
```

```
└ javax.swing.JComponent
```

```
└ javax.swing.JPanel
```

```
└ makourou.vsx.tools.ImageSplitPane
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible

```
public class ImageSplitPane  
extends javax.swing.JPanel
```

ImageSplitPane used to display two images in a split pane.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

javax.swing.JPanel.AccessibleJPanel

Nested classes/interfaces inherited from class javax.swing.JComponent

javax.swing.JComponent.AccessibleJComponent

Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent,
java.awt.Component.BaselineResizeBehavior,
java.awt.Component.BltBufferStrategy,
java.awt.Component.FlipBufferStrategy

Field Summary

static long [serialVersionUID](#)

Fields inherited from class javax.swing.JComponent

accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui,
UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT,
WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

[ImageSplitPane](#)(java.awt.image.BufferedImage image1, java.awt.image.BufferedImage image2)

The constructor of the class.

Method Summary

javax.swing.JSplitPane [getSplitPane](#)()

This method is used to return the split pane with the images.

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation,


```
getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getVerifyInputWhenFocusTarget,
getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY,
grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint,
isPaintingTile, isRequestFocusEnabled, isValidRoot, paint,
paintBorder, paintChildren, paintComponent, paintImmediately,
paintImmediately, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding,
processKeyEvent, processMouseEvent, processMouseMotionEvent,
putClientProperty, registerKeyboardAction, registerKeyboardAction,
removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus,
requestFocusInWindow, requestFocusInWindow, resetKeyboardActions,
reshape, revalidate, scrollRectToVisible, setActionMap,
setAlignmentX, setAlignmentY, setAutoscrolls, setBackground,
setBorder, setComponentPopupMenu, setDebugGraphicsOptions,
setDefaultLocale, setDoubleBuffered, setEnabled,
setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu,
setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPreferredSize,
setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction,
update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, addContainerListener, addImpl,
addPropertyChangeListener, addPropertyChangeListener,
applyComponentOrientation, areFocusTraversalKeysSet, countComponents,
deliverEvent, doLayout, findComponentAt, findComponentAt,
getComponent, getComponentAt, getComponentAt, getComponentCount,
getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider,
isFocusTraversalPolicySet, layout, list, list, locate, minimumSize,
paintComponents, preferredSize, printComponents,
processContainerEvent, processEvent, remove, remove,
removeContainerListener, setComponentZOrder, setFocusCycleRoot,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener,
addHierarchyBoundsListener, addHierarchyListener,
addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage,
checkImage, coalesceEvents, contains, createImage, createImage,
createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods,
firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange,
```

```
getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget,
getFocusCycleRootAncestor, getFocusListeners,
getFocusTraversalKeysEnabled, getFont, getForeground,
getGraphicsConfiguration, getHierarchyBoundsListeners,
getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners,
getName, getParent, getPeer, getPropertyChangeListeners,
getPropertyChangeListeners, getSize, getToolkit, getTreeLock,
gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside,
isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable,
isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet,
isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp,
list, list, list, location, lostFocus, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll,
postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent,
processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, resize,
resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled,
setIgnoreRepaint, setLocale, setLocation, setLocation, setName,
setSize, setSize, show, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait,
wait, wait
```

Field Detail

`serialVersionUID`

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

`ImageSplitPane`

```
public ImageSplitPane(java.awt.image.BufferedImage image1,
```

```
java.awt.image.BufferedImage image2)
```

The constructor of the class. Gets as input two buffered images.

Parameters:

image1 -

image2 -

Method Detail

getSplitPane

```
public javax.swing.JSplitPane getSplitPane()
```

This method is used to return the split pane with the images.

Returns:

a split pane.

makourou.vsx.algorithms

Class KMeansClustering

```
java.lang.Object
```

```
└─ makourou.vsx.algorithms.KMeansClustering
```

```
public class KMeansClustering
```

```
extends java.lang.Object
```

This class implements the K-Means Clustering algorithm.

Since:

1.0

Constructor Summary

```
KMeansClustering(java.awt.image.BufferedImage input)
```

The constructor of the class.

Method Summary

```
java.awt.image.BufferedImage getOutput()
```

Returns the output image.

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait
```

Constructor Detail

KMeansClustering

```
public KMeansClustering(java.awt.image.BufferedImage input)
```

The constructor of the class. Sets the input image.

Parameters:

input - the input image.

Method Detail

getOutput

```
public java.awt.image.BufferedImage getOutput()
```

Returns the output image.

Returns:

the image after the processing.

makourou.vsx.tools

Class MediaPlayer

```
java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └── makourou.vsx.tools.MediaPlayer
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.media.ControllerListener

```
public class MediaPlayer
extends javax.swing.JPanel
implements javax.media.ControllerListener
```

The MediaPlayer class is used to create a new Player object. The player added onto a JPanel. A FrameGrabbingControl used to take screenshots from the player and a FramePositioningControl used to set video stream to a specific frame.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JPanel

javax.swing.JPanel.AccessibleJPanel

Nested classes/interfaces inherited from class javax.swing.JComponent

javax.swing.JComponent.AccessibleJComponent

Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent,
java.awt.Component.BaselineResizeBehavior,
java.awt.Component.BltBufferStrategy,
java.awt.Component.FlipBufferStrategy

Field Summary

static long [serialVersionUID](#)

Fields inherited from class javax.swing.JComponent

accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui,
UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT,
WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT,
TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

[MediaPanel](#)(java.net.URL mediaURL)
Creates new media panel.

Method Summary

void	controllerUpdate (javax.media.ControllerEvent evt) This method is called when an event is generated.
VideoFrame	getFrame () Grab a frame from the video.
static void	setFrame (int frameNumber) Sets the video stream to a specific frame.
boolean	stopPlayer () Stops the player.

Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus,

```
isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint,
isPaintingTile, isRequestFocusEnabled, isValidRoot, paint,
paintBorder, paintChildren, paintComponent, paintImmediately,
paintImmediately, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding,
processKeyEvent, processMouseEvent, processMouseMotionEvent,
putClientProperty, registerKeyboardAction, registerKeyboardAction,
removeAncestorListener, removeNotify, removeVetoableChangeListener,
repaint, repaint, requestDefaultFocus, requestFocus, requestFocus,
requestFocusInWindow, requestFocusInWindow, resetKeyboardActions,
reshape, revalidate, scrollRectToVisible, setActionMap,
setAlignmentX, setAlignmentY, setAutoscrolls, setBackground,
setBorder, setComponentPopupMenu, setDebugGraphicsOptions,
setDefaultLocale, setDoubleBuffered, setEnabled,
setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu,
setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPreferredSize,
setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI,
setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction,
update
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener, addImpl,
addPropertyChangeListener, addPropertyChangeListener,
applyComponentOrientation, areFocusTraversalKeysSet, countComponents,
deliverEvent, doLayout, findComponentAt, findComponentAt,
getComponent, getComponentAt, getComponentAt, getComponentCount,
getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider,
isFocusTraversalPolicySet, layout, list, list, locate, minimumSize,
paintComponents, preferredSize, printComponents,
processContainerEvent, processEvent, remove, remove, removeAll,
removeContainerListener, setComponentZOrder, setFocusCycleRoot,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusBackward, transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener,
addHierarchyBoundsListener, addHierarchyListener,
addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage,
checkImage, coalesceEvents, contains, createImage, createImage,
createVolatileImage, createVolatileImage, disableEvents,
dispatchEvent, enable, enableEvents, enableInputMethods,
firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange,
getBackground, getBounds, getColorModel, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget,
getFocusCycleRootAncestor, getFocusListeners,
getFocusTraversalKeysEnabled, getFont, getForeground,
```

```
getGraphicsConfiguration, getHierarchyBoundsListeners,
getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocale, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners,
getName, getParent, getPeer, getPropertyChangeListeners,
getPropertyChangeListeners, getSize, getToolkit, getTreeLock,
gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside,
isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable,
isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet,
isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp,
list, list, list, location, lostFocus, mouseDown, mouseDrag,
mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll,
postEvent, prepareImage, prepareImage, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent,
processMouseWheelEvent, remove, removeComponentListener,
removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, resize,
resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled,
setIgnoreRepaint, setLocale, setLocation, setLocation, setName,
setSize, setSize, show, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait,
wait, wait
```

Field Detail

serialVersionUID

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

MediaPanel

```
public MediaPanel(java.net.URL mediaURL)
```

Creates new media panel.

Parameters:

mediaURL - media file location.

Method Detail

controllerUpdate

```
public void controllerUpdate(javax.media.ControllerEvent evt)
```

This method is called when an event is generated.

Specified by:

`controllerUpdate` in interface `javax.media.ControllerListener`

Parameters:

`evt` - The event generated.

getFrame

```
public VideoFrame getFrame()
```

Grab a frame from the video. Converts the Buffer object to Image and then to BufferedImage.

Returns:

the frame as a buffered image

stopPlayer

```
public boolean stopPlayer()
```

Stops the player.

Returns:

true

setFrame

```
public static void setFrame(int frameNumber)
```

Sets the video stream to a specific frame.

`makourou.vsx.algorithms`

Class Morphology

`java.lang.Object`

└─ `makourou.vsx.algorithms.Morphology`

```
public class Morphology  
extends java.lang.Object
```

This class implements the erosion and dilation operations. These operations can be applied to binary images.

Since:

1.0

Constructor Summary

[Morphology](#) ()

Method Summary

java.awt.image.BufferedImage	PerformDilate (java.awt.image.BufferedImage input) This method performs the dilate operation.
java.awt.image.BufferedImage	PerformErode (java.awt.image.BufferedImage input) This method performs the erode operation.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Morphology

```
public Morphology ()
```

Method Detail

PerformDilate

```
public java.awt.image.BufferedImage  
PerformDilate(java.awt.image.BufferedImage input)
```

This method performs the dilate operation.

Parameters:

input - the source image

Returns:

the image after dilation

PerformErode

```
public java.awt.image.BufferedImage  
PerformErode(java.awt.image.BufferedImage input)
```

This method performs the erode operation.

Parameters:

input - the source image

Returns:

the image after erosion

makourou.vsx.algorithms

Class ObjectIsolation

java.lang.Object

└─ **makourou.vsx.algorithms.ObjectIsolation**

```
public class ObjectIsolation
extends java.lang.Object
```

This class implements the Object Isolation algorithm. The algorithm is used to choose the largest object of a binary image and to erase the other smaller objects.

Since:

1.0

Constructor Summary

[ObjectIsolation](#)(java.awt.image.BufferedImage input)

The constructor of the class.

Method Summary

java.awt.image.BufferedImage [getOutput](#)()

Returns the output image.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ObjectIsolation

```
public ObjectIsolation(java.awt.image.BufferedImage input)
```

The constructor of the class. Sets the input image.

Parameters:

input - the input image.

Method Detail

getOutput

```
public java.awt.image.BufferedImage getOutput()
```

Returns the output image.

Returns:

the image after the processing.

makourou.vsx.algorithms

Class OutlineExtractor

java.lang.Object

└─ **makourou.vsx.algorithms.OutlineExtractor**

```
public class OutlineExtractor
```

```
extends java.lang.Object
```

This class implements the Outline Extractor algorithm. The algorithm is used to draw the outline of a binary image with one object.

Since:

1.0

Constructor Summary

[OutlineExtractor](#)(java.awt.image.BufferedImage input)

The constructor of the class.

Method Summary

java.awt.image.BufferedImage	getOutput () Returns the output image.
java.util.Vector<java.lang.Float>	getPoints () Returns the points of the outline.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

OutlineExtractor

```
public OutlineExtractor(java.awt.image.BufferedImage input)
```

The constructor of the class. Sets the input image.

Parameters:

input - the input image

Method Detail

getPoints

```
public java.util.Vector<java.lang.Float> getPoints()
```

Returns the points of the outline.

Returns:

outline points.

getOutput

```
public java.awt.image.BufferedImage getOutput()
```

Returns the output image.

Returns:

the image after the processing.

makourou.vsx.tools

Class ResultsFrame

```
java.lang.Object
```

```
└─ java.awt.Component
```

```
    └─ java.awt.Container
```

```
        └─ java.awt.Window
```

```
            └─ java.awt.Frame
```

```
                └─ javax.swing.JFrame
```

```
                    └─ makourou.vsx.tools.ResultsFrame
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class ResultsFrame  
extends javax.swing.JFrame
```

The ResultsFrame is used to display the similarity results. The results displayed in a JList.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

`javax.swing.JFrame.AccessibleJFrame`

Nested classes/interfaces inherited from class java.awt.Frame

`java.awt.Frame.AccessibleAWTFrame`

Nested classes/interfaces inherited from class java.awt.Window

`java.awt.Window.AccessibleAWTWindow`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent,`
`java.awt.Component.BaselineResizeBehavior,`
`java.awt.Component.BltBufferStrategy,`
`java.awt.Component.FlipBufferStrategy`

Field Summary

static long [serialVersionUID](#)

Fields inherited from class javax.swing.JFrame

accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.awt.Frame

CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

[ResultsFrame](#) ()

Creates new form ResultsFrame

Method Summary

void [setResults](#) (java.lang.String[] results)

This method is called to set similarity results into results list.

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics,

getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOwnedWindows, getOwner, getOwnerlessWindows, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImages, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setSize, setSize, setVisible, show, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove,


```
removeAll, removeContainerListener, setComponentZOrder,
setFocusTraversalKeys, setFocusTraversalPolicy,
setFocusTraversalPolicyProvider, setFont, transferFocusBackward,
transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener,
addHierarchyBoundsListener, addHierarchyListener,
addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage,
checkImage, coalesceEvents, contains, contains, createImage,
createImage, createVolatileImage, createVolatileImage, disable,
disableEvents, dispatchEvent, enable, enable, enableEvents,
enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBaseline,
getBaselineResizeBehavior, getBounds, getBounds, getColorModel,
getComponentListeners, getComponentOrientation, getCursor,
getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled,
getFont, getFontMetrics, getForeground, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint,
getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocation, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners,
getName, getParent, getPeer, getPropertyChangeListeners,
getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth,
getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside,
isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp,
list, list, list, location, lostFocus, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll,
prepareImage, prepareImage, printAll, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent, processKeyEvent,
processMouseEvent, processMouseMotionEvent, processMouseWheelEvent,
removeComponentListener, removeFocusListener,
removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener,
removeMouseMotionListener, removeMouseWheelListener,
removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resize, resize, setBackground,
setComponentOrientation, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint,
setLocale, setLocale, setLocation, setMaximumSize, setName,
setPreferredSize, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Methods inherited from interface java.awt.MenuContainer

```
getFont, postEvent
```

Field Detail

serialVersionUID

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

ResultsFrame

```
public ResultsFrame()
```

Creates new form ResultsFrame

Method Detail

setResults

```
public void setResults(java.lang.String[] results)
```

This method is called to set similarity results into results list.

Parameters:

results -

makourou.vsx

Class TestAlgorithms

```
java.lang.Object
```

```
└─ makourou.vsx.TestAlgorithms
```

```
public class TestAlgorithms
```

```
extends java.lang.Object
```

This method executes the algorithms which are used to export the outline and returns the results in each phase. This Class is used only for testing purposes and it is not part of the main system.

Since:
1.0

Constructor Summary	
TestAlgorithms()	

Method Summary	
static void	main (java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

TestAlgorithms

```
public TestAlgorithms()
```

Method Detail

main

```
public static void main(java.lang.String[] args)  
    The main method.  
Parameters:  
    arg[0] - a picture's file name.  
Since:  
    1.0
```

makourou.vsx.tools
Class VideoFrame

```
java.lang.Object  
└─ makourou.vsx.tools.VideoFrame
```

```
public class VideoFrame
extends java.lang.Object
```

The `VideoFrame` class is used in order to store a frame number and the frame that corresponds in this number.

Since:

1.0

Field Summary	
java.awt.image.BufferedImage	frameImage
int	frameNumber

Constructor Summary	
VideoFrame()	

Method Summary	
java.awt.image.BufferedImage	getFrameImage() Gets a frame
int	getFrameNumber() Gets a frame number
void	setFrameImage(java.awt.image.BufferedImage frameImage) Sets a frame.
void	setFrameNumber(int frameNumber) Sets a frame number.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

frameNumber

```
public int frameNumber
```

frameImage

```
public java.awt.image.BufferedImage frameImage
```

Constructor Detail

VideoFrame

```
public VideoFrame()
```

Method Detail

setFrameNumber

```
public void setFrameNumber(int frameNumber)
```

Sets a frame number.

Parameters:

frameNumber -

setFrameImage

```
public void setFrameImage(java.awt.image.BufferedImage frameImage)
```

Sets a frame.

Parameters:

frameImage -

getFrameNumber

```
public int getFrameNumber()
```

Gets a frame number

Returns:

an integer corresponding to a frame.

getFrameImage

```
public java.awt.image.BufferedImage getFrameImage()
```

Gets a frame

Returns:

a BufferedImage corresponding to a frame.

makourou.vsx

Class VideoShapeExtractor

```

java.lang.Object
├─ java.awt.Component
│   └─ java.awt.Container
│       └─ java.awt.Window
│           └─ java.awt.Frame
│               └─ javax.swing.JFrame
│                   └─ makourou.vsx.VideoShapeExtractor

```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

```

public class VideoShapeExtractor
extends javax.swing.JFrame

```

VideoShapeExtractor is the main class of the application. It implements the GUI and performs all the operations to extract the shape from a video.

Since:

1.0

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

javax.swing.JFrame.AccessibleJFrame

Nested classes/interfaces inherited from class java.awt.Frame

java.awt.Frame.AccessibleAWTFrame

Nested classes/interfaces inherited from class java.awt.Window

java.awt.Window.AccessibleAWTWindow

Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent,
java.awt.Component.BaselineResizeBehavior,
java.awt.Component.BltBufferStrategy,
java.awt.Component.FlipBufferStrategy

Field Summary

static long	<u>serialVersionUID</u>
-------------	---

Fields inherited from class javax.swing.JFrame

accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.awt.Frame

CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

[VideoShapeExtractor](#) ()

Creates new form VideoShapeExtractor

Method Summary

static void [appendInfoTextArea](#) (java.lang.String text)

This method is called from other classes in order to put information into frame's text area.

static void [main](#) (java.lang.String[] args)

The main method.

static void [setVideoFrame](#) (int frameNumber)

This method is called to set the video stream to a specific frame number.

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOwnedWindows, getOwner,


```
getOwnerlessWindows, getToolkit, getWarningString,  
getWindowFocusListeners, getWindowListeners, getWindows,  
getWindowStateListeners, hide, isActive, isAlwaysOnTop,  
isAlwaysOnTopSupported, isFocusableWindow, isFocusCycleRoot,  
isFocused, isLocationByPlatform, isShowing, pack, postEvent,  
processEvent, processWindowFocusEvent, processWindowStateEvent,  
removeWindowFocusListener, removeWindowListener,  
removeWindowStateListener, reshape, setAlwaysOnTop, setBounds,  
setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot,  
setIconImages, setLocationByPlatform, setLocationRelativeTo,  
setMinimumSize, setModalExclusionType, setSize, setSize, setVisible,  
show, toBack, toFront
```

Methods inherited from class java.awt.Container

```
add, add, add, add, add, add, addContainerListener,  
applyComponentOrientation, areFocusTraversalKeysSet, countComponents,  
deliverEvent, doLayout, findComponentAt, findComponentAt,  
getAlignmentX, getAlignmentY, getComponent, getComponentAt,  
getComponentAt, getComponentCount, getComponents, getComponentZOrder,  
getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout,  
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize,  
insets, invalidate, isAncestorOf, isFocusCycleRoot,  
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout,  
list, list, locate, minimumSize, paint, paintComponents,  
preferredSize, print, printComponents, processContainerEvent, remove,  
removeAll, removeContainerListener, setComponentZOrder,  
setFocusTraversalKeys, setFocusTraversalPolicy,  
setFocusTraversalPolicyProvider, setFont, transferFocusBackward,  
transferFocusDownCycle, validate, validateTree
```

Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener,  
addHierarchyBoundsListener, addHierarchyListener,  
addInputMethodListener, addKeyListener, addMouseListener,  
addMouseMotionListener, addMouseWheelListener, bounds, checkImage,  
checkImage, coalesceEvents, contains, contains, createImage,  
createImage, createVolatileImage, createVolatileImage, disable,  
disableEvents, dispatchEvent, enable, enable, enableEvents,  
enableInputMethods, firePropertyChange, firePropertyChange,  
firePropertyChange, firePropertyChange, firePropertyChange,  
firePropertyChange, firePropertyChange, firePropertyChange,  
firePropertyChange, getBackground, getBaseline,  
getBaselineResizeBehavior, getBounds, getBounds, getColorModel,  
getComponentListeners, getComponentOrientation, getCursor,  
getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled,  
getFont, getFontMetrics, getForeground, getHeight,  
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint,  
getInputMethodListeners, getInputMethodRequests, getKeyListener,  
getLocation, getLocation, getLocationOnScreen, getMouseListeners,  
getMouseMotionListeners, getMousePosition, getMouseWheelListeners,  
getName, getParent, getPeer, getPropertyChangeListeners,  
getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth,
```

```
getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside,
isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp,
list, list, list, location, lostFocus, mouseDown, mouseDrag,
mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll,
prepareImage, prepareImage, printAll, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent, processKeyEvent,
processMouseEvent, processMouseMotionEvent, processMouseWheelEvent,
removeComponentListener, removeFocusListener,
removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener,
removeMouseMotionListener, removeMouseWheelListener,
removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resize, resize, setBackground,
setComponentOrientation, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint,
setLocale, setLocation, setLocation, setMaximumSize, setName,
setPreferredSize, show, size, toString, transferFocus,
transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait,
wait, wait
```

Methods inherited from interface java.awt.MenuContainer

```
getFont, postEvent
```

Field Detail

serialVersionUID

```
public static final long serialVersionUID
```

See Also:

[Constant Field Values](#)

Constructor Detail

VideoShapeExtractor

```
public VideoShapeExtractor()
    Creates new form VideoShapeExtractor
```

Method Detail

main

```
public static void main(java.lang.String[] args)
```

The main method.

Parameters:

args - the command line arguments

Since:

1.0

appendInfoTextArea

```
public static void appendInfoTextArea(java.lang.String text)
```

This method is called from other classes in order to put information into frame's text area.

Parameters:

text - the text argument

Since:

1.0

setVideoFrame

```
public static void setVideoFrame(int frameNumber)
```

This method is called to set the video stream to a specific frame number.

Parameters:

frameNumber -

makourou.vsx.tools

Class XmlPdbWriter

```
java.lang.Object
```

```
└─ makourou.vsx.tools.XmlPdbWriter
```

```
public class XmlPdbWriter
```

```
extends java.lang.Object
```

XmlPdbWriter is used to create an XML file which includes the points of the outlines.

Since:

1.0

Constructor Summary

```
XmlPdbWriter()
```

Creates new XmlPdbWriter.

Method Summary

static void [writePdb](#)(java.util.Vector<java.util.Vector<java.lang.Float>> vectorsOfPo:
This method is called to write the XML file.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

XmlPdbWriter

```
public XmlPdbWriter()  
    Creates new XmlPdbWriter.
```

Method Detail

writePdb

```
public static void  
writePdb(java.util.Vector<java.util.Vector<java.lang.Float>> vectorsO  
fPoints)
```

This method is called to write the XML file.

Parameters:

vectorsOfPoints - the points of the outlines.

Αναφορές

- [1] Linda G. Shapiro, George C. Stockman, “*Computer Vision*”, Prentice-Hall 2001, ISBN 0-13-030796-3.
- [2] Abdollah Chalechale, Golshah Naghdy, Alfred Mertins, “*Sketch-based image matching Using Angular partitioning*”. IEEE Transactions on Systems, Man, and Cybernetics, Part A 35(1): 28-41 (2005)
- [3] B. G. Prasad, S. K. Gupta, K. K. Biswas: “*Color and Shape Index for Region-Based Image Retrieval*”. IWVF 2001: 716-728
- [4] Canny, J., “*A computational approach to edge detection*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 8 (1986), pp. 679-698.
- [5] Dzung L. Pham, Chenyang Xu, Jerry L. Prince, “*A survey of current methods in medical image segmentation*”. Annual Review of Biomedical Engineering Vol. 2: 315-337 (August 2000).
- [6] Edoardo Ardizzone, Marco La Cascia. “*Automatic Video Database Indexing and Retrieval*”. Multimedia Tools Appl. 4(1): 29-56 (1997).
- [7] Ioannis Andreou, Nikitas M. Sgouros: “*Computing, explaining and visualizing shape similarity in content-based image retrieval*”. Inf. Process. Manage. 41(5): 1121-1139 (2005).
- [8] J. B. MacQueen, “*Some Methods for classification and Analysis of Multivariate Observations*”, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297 (1967).
- [9] Jan Klima, Tomas Skopal, “*Shape Extraction Framework for Similarity Search in Image Databases*”, Dateso 2007, pp. 89–102, ISBN 80-7378-002-X.
- [10] Margaret M. Fleck, David A. Forsyth, Christoph Bregler: “*Finding Naked People*”. ECCV (2) 1996: 593-602
- [11] Richard Nock, Frank Nielsen, “*Statistical Region Merging*”. IEEE transactions on pattern analysis and machine intelligence, vol. 26, no. 11, November 2004.
- [12] Stéphane Marchand-Maillet, “*Content-Based Video Retrieval: An Overview.*” Tech. Rep. 00.06, CUI - University of Geneva, Geneva, Switzerland, 2000.

- [13] Vitorino Ramos, Fernando Muge, “*Image Colour Segmentation by Genetic Algorithms*”. 11th Portuguese Conf. on Pattern Recognition, ISBN 972-96883-2-5, pp. 125-129 (May 11-12, 2000).
- [14] Wayne Niblack, Ron Barber, William Equitz, Myron Flickner, Eduardo H. Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, Gabriel Taubin, “*The QBIC Project: Querying Images by Content, Using Color, Texture, and Shape*”. Storage and Retrieval for Image and Video Databases (SPIE) 1993: 173-187
- [15] Xiang Sean Zhou, Thomas S. Huang, “*Edge-based structural features for content-based image retrieval*”. Pattern Recognition Letters 22(5) (2001), pp. 457-468.
- [16] Zujun Hou, T. S. Koh, “*Robust edge detection*”. Pattern Recognition 36(9) (2003), 2083-2091.
- [17] <http://java.sun.com/products/java-media/jmf/2.1.1/apidocs/>
- [18] <http://java.sun.com/products/java-media/jmf/1.0/guide/index.html>
- [19] <http://www.onjava.com/pub/a/onjava/2005/06/01/kgpjava.html?page=3>
- [20] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>
- [21] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>
- [22] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>
- [23] <https://jaistuff.dev.java.net/algorithms.html>
- [24] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html
- [25] <http://pro.corbis.com>

