



**ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ**

**ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ, ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

## **Πτυχιακή Εργασία**

**ΑΝΑΓΝΩΡΙΣΗ ΑΝΘΡΩΠΙΝΟΥ ΒΑΔΙΣΜΑΤΟΣ ΜΕ ΧΡΗΣΗ ΔΙΑΦΟΡΙΚΩΝ  
ΕΙΚΟΝΩΝ**

Όνοματεπώνυμο: Ουρουμίδου Ίρις

ΑΕΜ: 3272

Επιβλέπων: Νικολαΐδης Αθανάσιος, Αναπληρωτής Καθηγητής

**ΣΕΡΡΕΣ 2021**

**Υπεύθυνη Δήλωση** : Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδας.

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Νικολαΐδη Αθανάσιο, για τη δυνατότητα που μου έδωσε να ασχοληθώ με την παρούσα πτυχιακή εργασία και με αυτόν τον τρόπο να διευρύνω τους ορίζοντες μου.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξη που μου προσέφεραν καθ' όλη την διάρκεια των σπουδών μου και τους φίλους για τις αξέχαστες αναμνήσεις.

## ΠΕΡΙΛΗΨΗ

Η αναγνώριση βαδίσματος αναφέρεται συνήθως για να δηλώσει την ανθρώπινη ταυτοποίηση από το στυλ ή τον τρόπο με τον οποίο οι άνθρωποι περπατούν σε ακολουθίες εικόνων. Η διαφορά μεταξύ γειτονικών καρέ του ανθρώπινου βαδίσματος περιέχει χρήσιμες πληροφορίες για την αναγνώριση του ανθρώπινου βαδίσματος. Βασισμένη στην προηγούμενη ιδέα, μελετάται και υλοποιείται σε αυτήν την εργασία μια μέθοδος αναγνώρισης ανθρώπινου βαδίσματος, η οποία βασίζεται στη μέση διαφορική εικόνα βαδίσματος (Average Gait Differential Image-AGDI). Η AGDI δημιουργείται από τη συσσώρευση της διαφοράς σιλουετών μεταξύ γειτονικών καρέ. Το πλεονέκτημα αυτής της μεθόδου έγκειται στο ότι, ως εικόνα χαρακτηριστικών, η AGDI μπορεί να διατηρήσει τόσο την κινητική όσο και τη στατική πληροφορία του περπατήματος. Η δισδιάστατη ανάλυση κυρίων συνιστωσών (2DPCA- Two-dimensional principal component analysis) χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από την AGDI. Η τεχνική αυτή εφαρμόζεται σε υπάρχουσες βάσεις δεδομένων προκειμένου να επαληθευτεί η αποδοτικότητά της.

## ABSTRACT

Gait recognition is a term that usually refers to human identification by the style/way people walk in image sequences. The difference between adjacent frames of human walking contains useful information for human gait identification. Based on the previous idea, a silhouettes difference based human gait recognition method based on the average gait differential image (AGDI) is being studied and implemented in this thesis. The AGDI is generated by the accumulation of the silhouettes difference between adjacent frames. The advantage of this method lies in that, as a feature image, it can preserve both the kinetic and static information of walking. Two-dimensional principal component analysis (2DPCA) is used to extract features from the AGDI. This technique is being applied to existing databases in order to verify its efficiency.

# ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	- 3 -
ΠΕΡΙΛΗΨΗ.....	- 4 -
ABSTRACT.....	- 5 -
ΠΕΡΙΕΧΟΜΕΝΑ.....	- 6 -
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	- 8 -
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	- 10 -
ΚΕΦΑΛΑΙΟ 1.....	- 11 -
Εισαγωγή.....	- 11 -
1.1 Γενικά.....	- 11 -
1.2 Σχετική δουλειά.....	- 12 -
1.3 Αντικείμενο Πτυχιακής Εργασίας.....	- 14 -
1.4 Διάρθρωση της Πτυχιακής Εργασίας.....	- 14 -
ΚΕΦΑΛΑΙΟ 2.....	- 16 -
Αλγόριθμος Αναγνώρισης Ανθρώπινου Βαδίσματος.....	- 16 -
2.1 Γενικά.....	- 16 -
2.2 Κατασκευή Μέσης Διαφορικής Εικόνας Βαδίσματος (AGDI).....	- 16 -
2.2.1 Τμηματοποίηση σιλουέτας.....	- 16 -
2.2.2 Κανονικοποίηση.....	- 16 -
2.2.3 Ευθυγράμμιση και Αφαίρεση.....	- 16 -
2.2.4 Κατασκευή της Μέσης Διαφορικής Εικόνας Βαδίσματος.....	- 18 -
2.3 Εξαγωγή Χαρακτηριστικών.....	- 19 -
2.4 Αναγνώριση Σιλουετών.....	- 22 -
2.5 Ανακατασκευή με βάση τη μέση διαφορική εικόνα βαδίσματος.....	- 22 -

ΚΕΦΑΛΑΙΟ 3 .....	- 24 -
Περιβάλλον Ανάπτυξης Λογισμικού Matlab.....	- 24 -
3.1 Τι είναι το Matlab; .....	- 24 -
3.2 Το γραφικό περιβάλλον χρήστη – GUI .....	- 25 -
ΚΕΦΑΛΑΙΟ 4 .....	- 26 -
Υλοποίηση Αναγνώρισης Ανθρώπινου Βαδίσματος.....	- 26 -
4.1 Γενικά.....	- 26 -
4.2 Εφαρμογή Αναγνώρισης Ανθρώπινου Βαδίσματος .....	- 31 -
ΚΕΦΑΛΑΙΟ 5 .....	- 46 -
5.1 Συμπεράσματα - Αποτελέσματα.....	- 46 -
5.2 Μελλοντικές Έρευνες και Επεκτάσεις του Προγράμματος.....	- 48 -
ΒΙΒΛΙΟΓΡΑΦΙΑ / ΑΝΑΦΟΡΕΣ.....	- 49 -
ΠΑΡΑΡΤΗΜΑ Α' .....	- 52 -
ΠΑΡΑΡΤΗΜΑ Β'.....	- 55 -
ΠΑΡΑΡΤΗΜΑ Γ' .....	- 69 -

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.2.3. 1 Το κέντρο μιας σιλουέτας .....	- 17 -
Εικόνα 2.2.3. 2 Διαφορικές εικόνες και μέση διαφορική εικόνα βαδίσματος .....	- 18 -
Εικόνα 2.3. 1 Οι ανακατασκευασμένες υπο-εικόνες ( $k= 1,2,3,5,10,20$ ) και οι αρχικές εικόνες .....	- 22 -
Εικόνα 4.1. 1 Σχεδιάγραμμα Αναγνώρισης .....	- 27 -
Εικόνα 4.1. 2 (α) Ακατέργαστη εικόνα , (β) Φόντο , (γ) Αφαίρεση Φόντου , .....	- 28 -
Εικόνα 4.1. 3 Αρχική Εικόνα, 0 μοίρες .....	- 28 -
Εικόνα 4.1. 4 Κανονικοποιημένη σιλουετα με το κέντρο της, 0 μοίρες .....	- 29 -
Εικόνα 4.1. 5 Διαφορικές εικόνες και μέση διαφορική εικόνα (AGDI), 90 μοίρες.....	- 29 -
Εικόνα 4.1. 6 Ανακατασκευή Εικόνας Ατόμου 90 μοιρών και αρχική εικόνα. ....	- 30 -
Εικόνα 4.2. 1 AGDI_GUI, Γραφικό περιβάλλον.....	- 31 -
Εικόνα 4.2. 2 AGDI_GUI, Βασικά Μέρη Εφαρμογής .....	- 32 -
Εικόνα 4.2. 3 AGDI_GUI, Περιγραφές Buttons.....	- 33 -
Εικόνα 4.2. 4 VIEW_IMAGES, Silhouette Viewer .....	- 34 -
Εικόνα 4.2. 5 Silhouette Viewer, Train/ Person 17/ Angle 90, Πρώτο Καρέ(61 συνολικά).....	- 35 -
Εικόνα 4.2. 6 Silhouette Viewer, Train/ Person 17/ Angle 90, Καρέ 28 .....	- 36 -
Εικόνα 4.2. 7 Calculation of AGDI .....	- 37 -
Εικόνα 4.2. 8 Calculation of AGDI, calculations in progress.....	- 38 -
Εικόνα 4.2. 9 Calculation of AGDI, Finished.....	- 39 -
Εικόνα 4.2. 10 Feature Extraction .....	- 40 -
Εικόνα 4.2. 11 Feature Extraction, calculations in progress .....	- 40 -
Εικόνα 4.2. 12 Feature Extraction, Finished.....	- 41 -
Εικόνα 4.2. 13 Choose a Person .....	- 42 -



Εικόνα 4.2. 14 Choose an Angle .....	- 42 -
Εικόνα 4.2. 15 Operation Completed .....	- 43 -
Εικόνα 4.2. 16 Person Recognized .....	- 44 -
Εικόνα 4.2. 17 Person was not Recognized .....	- 45 -
Εικόνα 5.1 1 Bar Graph Results .....	- 47 -

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 5.1. 1 Πίνακας Αποτελεσμάτων ..... - 46 -

Πίνακας Α. 1 Πληροφορίες Ατόμων Αναγνώρισης..... - 54 -

# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

### 1.1 Γενικά

Με την ανάπτυξη της πληροφορικής και της τεχνολογίας του Διαδικτύου, είναι απαραίτητο να ταυτοποιούμε και να εξουσιοδοτούμε τους ανθρώπους με ασφαλή τρόπο. Η ταχεία ανάπτυξη του ηλεκτρονικού εμπορίου, επίσης, χρειάζεται μια αξιόπιστη μέθοδο ταυτοποίησης για να εξασφαλίσει την ασφάλεια συναλλαγής. Ως μια πολλά υποσχόμενη μέθοδος ελέγχου ταυτότητας, η βιομετρική προσελκύει όλο και περισσότερη προσοχή. Η βιομετρική υπερνικά τις εγγενείς ατέλειες και περιορισμούς της συμβατικής τεχνολογίας αναγνώρισης και παρέχει μια εξαιρετικά ασφαλή μέθοδο αναγνώρισης και ταυτοποίησης. Οι παραδοσιακοί βιομετρικοί πόροι περιλαμβάνουν δακτυλικά αποτυπώματα, εικόνες προσώπου και ίριδας, που έχουν χρησιμοποιηθεί ευρέως για ταυτοποίηση. Ωστόσο, αυτά τα βιομετρικά χαρακτηριστικά έχουν τα ακόλουθα μειονεκτήματα:

A. Αυτά τα χαρακτηριστικά δε μπορούν να ληφθούν από μια σχετικά μακρινή απόσταση.

B. Η συνεργασία του χρήστη απαιτείται για να πάρουμε καλά αποτελέσματα. Ως νέα βιομετρική μέθοδος, η αναγνώριση του ανθρώπου με βάση το βάδισμα υπερνικά τον παραπάνω περιορισμό και προσελκύει όλο και περισσότερους ερευνητές.

Το ανθρώπινο βάδισμα, που είναι ο τρόπος που περπατά κάποιος, μελετήθηκε πρώτα στον τομέα της ιατρικής. Οι γιατροί ανέλυσαν το ανθρώπινο βάδισμα για να μάθουν αν οι ασθενείς είχαν προβλήματα υγείας [1,2]. Αργότερα, οι ερευνητές διαπίστωσαν ότι, όπως ακριβώς το δακτυλικό αποτύπωμα και η ίριδα, σχεδόν καθένας έχει το δικό του διακριτικό στυλ περπατήματος [3,4]. Έτσι, θεωρήθηκε ότι το βάδισμα μπορούσε επίσης να χρησιμοποιηθεί ως ένα βιολογικό χαρακτηριστικό για την αναγνώριση ατόμου. Παρά το γεγονός ότι επηρεάζεται από την παρουσία ρούχων, παπουτσιών, τη γωνία θέασης ή το περιβάλλον, το ανθρώπινο βάδισμα εξακολουθεί να είναι μια πολλά υποσχόμενη μέθοδος αναγνώρισης.

Το ανθρώπινο βάδισμα μπορεί να θεωρηθεί ως μια ακολουθία εικόνων. Ωστόσο, οι περισσότερες υπάρχουσες ανεξάρτητες από μοντέλα μέθοδοι αναγνώρισης που βασίζονται στο βάδισμα εξάγουν χαρακτηριστικά από μια ακολουθία εικόνων χωρίς να λαμβάνουν υπόψη την περιεχόμενη χωροχρονική πληροφορία. Η μέθοδος που μελετάται και υλοποιείται στην παρούσα πτυχιακή εργασία επικεντρώνεται στη διαφορά μεταξύ των εικόνων μιας ακολουθίας καθώς κατασκευάζει την εικόνα χαρακτηριστικών. Η διαδικασία μπορεί να περιγραφεί ως ακολούθως: Οι σιλουέτες κανονικοποιούνται στο ίδιο ύψος και ευθυγραμμίζονται βάσει του κέντρου βάρους. Μετά, η διαφορά μεταξύ δύο διαδοχικών σιλουετών συσσωρεύεται για να πάρουμε τη μέση διαφορική εικόνα βαδίσματος (AGDI) που χρησιμοποιείται ως η εικόνα χαρακτηριστικών κάποιου ατόμου που βαδίζει. Η διδιάστατη ανάλυση κυρίων συνιστωσών χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από την AGDI.

## 1.2 Σχετική δουλειά

Συνήθως, η αναγνώριση που βασίζεται στο ανθρώπινο βάδισμα περιλαμβάνει διάφορες προσεγγίσεις, όπως περπάτημα, τρέξιμο και πήδημα. Σε αυτήν την έρευνα θα περιορίσουμε την αναγνώριση στο βάδισμα. Επί του παρόντος, η αναγνώριση ανθρώπινου βαδίσματος μπορεί να χωριστεί σε δύο κατηγορίες: μέθοδοι που βασίζονται σε μοντέλα (model-based methods) και μέθοδοι που βασίζονται στην κίνηση (motion-based methods).

Οι προσεγγίσεις που βασίζονται σε μοντέλα έχουν στόχο να περιγράψουν την ανθρώπινη κίνηση χρησιμοποιώντας ένα μαθηματικό μοντέλο. Οι Cunado και λοιποί [5] χρησιμοποίησαν τον μετασχηματισμό Hough για να εξάγουν τις θέσεις των χεριών, ποδιών και του κορμού του σώματος και μετά χρησιμοποίησαν το αρθρωτό εκκρεμές για να αντιστοιχίσουν αυτά τα κινούμενα μέλη του σώματος. Οι Yoo και λοιποί [6] διαίρεσαν το σώμα σε κεφάλι, λαιμό, μέση, πόδια και χέρια με κατάτμηση εικόνας και μετά πήραν τις καμπύλες κίνησης από αυτά τα μέλη του σώματος, αντίστοιχα. Οι Lee και Grimson [7] χρησιμοποίησαν 7 ελλείψεις για να μοντελοποιήσουν το ανθρώπινο σώμα και τα χαρακτηριστικά κίνησης των ελλείψεων για την αναγνώριση ανθρώπου. Οι Yam και λοιποί [8,9] χρησιμοποίησαν δυναμικά συζευγμένους ταλαντωτές για να περιγράψουν και να αναλύσουν τον τρόπο βαδίσματος και τρεξίματος ενός ατόμου. Οι Tafazzoli και Safabakhsh [10] κατασκεύασαν μοντέλα κίνησης βασισμένα σε

ανατομικές αναλογίες. Στη συνέχεια ο μετασχηματισμός Fourier χρησιμοποιήθηκε για την ανάλυση του ανθρώπινου βαδίσματος.

Οι ελεύθερες μοντέλου μέθοδοι εστιάζουν στη στατιστική πληροφορία που εξάγεται από το ανθρώπινο βάδισμα. Οι Cheng και λοιποί [11] πήραν το μοντέλο HMM (Hidden Markov Model-Κρυμμένο Μαρκοβιανό Μοντέλο) για να αναλύσουν την σχέση ανάμεσα στον άνθρωπο και τις εικόνες βαδίσματός του. Οι Chen και λοιποί [12] χρησιμοποίησαν το παράλληλο HMM για να περιγράψουν τα χαρακτηριστικά του ανθρώπινου βαδίσματος. Οι Kale και λοιποί [13] χρησιμοποίησαν μοτίβα «τοιχογραφίας» για να πάρουν χαρακτηριστικά από μια ακολουθία εικόνων και τα χρησιμοποίησαν για την αναγνώριση ανθρώπου. Οι Murase και Sakai [14] επιτάχυναν τη σύγκριση του ανθρώπινου βαδίσματος με παραμετρική αναπαράσταση ιδιοχώρου. Οι Little και Boyd [15] εξήγαγαν βαθμωτά χαρακτηριστικά ανεξάρτητα από κλίμακα από την πληροφορία οπτικής ροής των βαδίζόντων σιλουετών για την αναγνώριση ατόμων. Οι Wang και λοιποί [16] εξήγαγαν χαρακτηριστικά από το «ξετύλιγμα» του εξωτερικού περιγράμματος της σιλουέτας και χρησιμοποίησαν PCA για την μείωση των διαστάσεων των χαρακτηριστικών. Οι Lee και λοιποί [17] υιοθέτησαν το γινόμενο των συντελεστών Fourier σαν μέτρο απόστασης μεταξύ περιγραμμάτων για να αναγνωρίσουν βάδισμα. Ο Hu [18] συνδύασε τη βελτιωμένη αναπαράσταση (Enhanced Gabor) της εικόνας ενέργειας βαδίσματος και την κανονικοποιημένη διακριτική ανάλυση τοπικού τανυστή (Regularized local tensor discriminate analysis) στην αναγνώριση ανθρώπου. Οι Hong και λοιποί [19] πρότειναν ένα πιθανοτικό πλαίσιο για την αναγνώριση ανθρώπου. Οι Wang και λοιποί [20] πρότειναν χωροχρονική ανάλυση πληροφορίας για την εξαγωγή χαρακτηριστικών. Οι Collins και λοιποί [21] εξάγουν πλαίσια-κλειδιά από την ακολουθία εικόνων και συγκρίνουν την ομοιότητά τους με κανονικοποιημένη συσχέτιση. Οι Sarkar και λοιποί [22] εκτίμησαν την ομοιότητα ανάμεσα στην ακολουθία εικόνων της συλλογής και την ακολουθία εικόνων δοκιμής υπολογίζοντας κατευθείαν τη συσχέτιση ανάμεσα στα ζευγάρια των πλαισίων. Ο Chen [23] πρότεινε μια μέθοδο αναγνώρισης ανθρώπου βασισμένη στη συσχέτιση εικόνας.

Η μέθοδος που υλοποιήθηκε στα πλαίσια της πτυχιακής εργασίας είναι παρόμοια με την εικόνα ενέργειας βαδίσματος (GEI=gait energy image), που προτάθηκε από τους Yu και λοιπούς [24], Han και Bhanu [25] και με την εικόνα ενέργειας διαφοράς πλαισίων (FDEI=frame difference

energy image) που προτάθηκε από τους Chen και λοιπούς [26]. Η βασική διαφορά έγκειται στην προσέγγιση για την δημιουργία της εικόνας χαρακτηριστικών. Η GEI λαμβάνεται αθροίζοντας απευθείας κάθε κανονικοποιημένη σιλουέτα. Η FDEI λαμβάνεται από την διαφορά κάθε δύο γειτονικών πλαισίων και συνδυάζοντάς την με την «αποθορυβοποιημένη» GEI. Η διαφορά μεταξύ κάθε δύο πλαισίων συσσωρεύεται για την δημιουργία της μέσης διαφορικής εικόνας βαδίσματος. Επίσης, βελτιώθηκε η μέθοδος εξαγωγής χαρακτηριστικών χρησιμοποιώντας την 2DPCA, που έχει χρησιμοποιηθεί στην εφαρμογή αναγνώρισης προσώπου [27].

Σε σύγκριση με τις πιο πρόσφατες εξελίξεις στον τομέα, οι συνεισφορές της παρούσας μεθόδου είναι οι εξής:

Μέθοδος Αναπαράστασης Βαδίσματος (Gait Representation Method): Προτάθηκε μια νέα αναπαράσταση χαρακτηριστικού βαδίσματος η οποία ονομάζεται Μέσης Διαφορικής Εικόνας Βαδίσματος (Average Gait Differencial Image). Συγκρίνοντάς την με την GEI, η μέθοδος που υλοποιείται έχει το πλεονέκτημα της καλύτερης απόδοσης.

Μέθοδος Εξαγωγής Χαρακτηριστικών (Feature Extraction Method): Η διδιάστατη ανάλυση κυρίων συνιστωσών (2DPCA= Two dimensional principal component analysis) χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από την AGDI (Average gait differencial image). Αυτή μπορεί να είναι πιο αποδοτική και να εξοικονομήσει περισσότερο χώρο αποθήκευσης σε σύγκριση με την ευρέως χρησιμοποιούμενη μονοδιάστατη ανάλυση κυρίων συνιστωσών (PCA).

### 1.3 Αντικείμενο Πτυχιακής Εργασίας

Οι στόχοι της εργασίας αυτής είναι η ανάπτυξη, η εφαρμογή και αξιολόγηση των αλγοριθμικών μεθόδων που αφορούν τον εντοπισμό και αναγνώριση ανθρώπινου βαδίσματος σε έτοιμες βάσεις δεδομένων και η υλοποίησή τους με τη βοήθεια των δυνατοτήτων που μας παρέχει το προγραμματιστικό περιβάλλον της Matlab.

### 1.4 Διάρθρωση της Πτυχιακής Εργασίας

Η παρούσα εργασία οργανώνεται ως εξής: Στο Κεφάλαιο 2 παρουσιάζεται ο μαθηματικός αλγόριθμος για την αναγνώριση ανθρώπινου βασίσματος και στο Κεφάλαιο 3 γίνεται μια εισαγωγή για το περιβάλλον της Matlab. Στο κεφάλαιο 4 γίνεται η υλοποίηση του αλγορίθμου αναγνώρισης με τη χρήση του περιβάλλοντος της Matlab. Τέλος, στο Κεφάλαιο 5

συγκεντρώνονται τα αποτελέσματα των πειραμάτων και τίθενται οι κατευθύνσεις για περαιτέρω έρευνα.

## ΚΕΦΑΛΑΙΟ 2

### Αλγόριθμος Αναγνώρισης Ανθρώπινου Βαδίσματος

#### 2.1 Γενικά

Σε αυτό το κεφάλαιο θα αναλύσουμε τον αλγόριθμο που χρησιμοποιούμε για την υλοποίηση της αναγνώρισης βαδίσματος και τα βήματα που πρέπει να κάνουμε ώστε να έχουμε τα επιθυμητά αποτελέσματα.

#### 2.2 Κατασκευή Μέσης Διαφορικής Εικόνας Βαδίσματος (AGDI)

##### 2.2.1 Τμηματοποίηση σιλουέτας

Το μοντέλο Gauss χρησιμοποιείται για να δημιουργηθεί το μοντέλο του φόντου από την αρχική ακολουθία των εικόνων. Για να αφαιρεθεί ο θόρυβος, σε κάθε εικόνα εφαρμόζεται το φίλτρο Gauss που τη θολώνει. Η μέθοδος που πρότειναν οι Wang και λοιποί [16] χρησιμοποιείται για να εξάγει το μοντέλο από ένα αντικείμενο που βαδίζει με βάση τις εικόνες.

##### 2.2.2 Κανονικοποίηση

Για να εξαλειφθεί το πρόβλημα της σχετικής μετατόπισης της μιας εικόνας από την άλλη, η κάθε σιλουέτα κανονικοποιείται στο ίδιο ύψος χρησιμοποιώντας την μέθοδο της bicubic interpolation (δικυβική παρεμβολή).

##### 2.2.3 Ευθυγράμμιση και Αφαίρεση

Καθορίζουμε το κέντρο της ισορροπίας  $(x_c, y_c)$ , της σιλουέτας ως εξής:



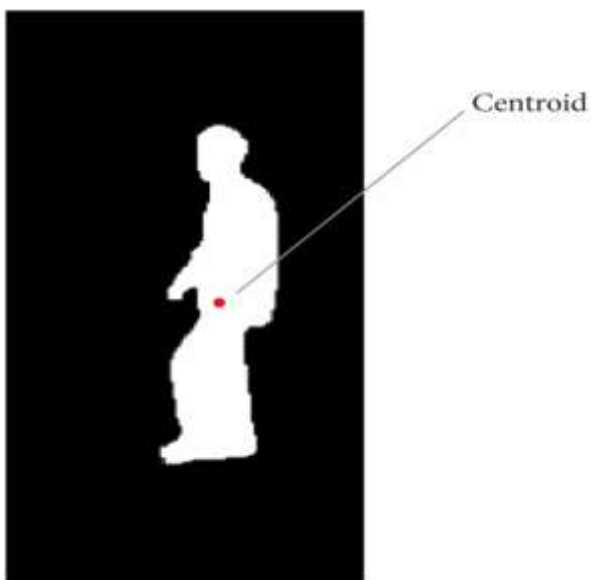
$$x_c = \frac{1}{n} \sum_{i=1}^n x_i$$

(1)

$$y_c = \frac{1}{n} \sum_{i=1}^n y_i$$

Όπου  $n$  ο αριθμός των εικονοστοιχείων στη σιλουέτα.

Η εικόνα 2.2.3.1, αμέσως παρακάτω, δείχνει πώς φαίνεται το κέντρο της ισορροπίας μιας σιλουέτας.



Εικόνα 2.2.3. 1 Το κέντρο μιας σιλουέτας

Υποθέτουμε πως οι  $I_j$  και  $I_{j+1}$  είναι δύο γειτονικές εικόνες ευθυγραμμισμένες με βάση το κέντρο της ισορροπίας τους, οπότε, η διαφορική εικόνα βαδίσματος  $D_j$  μπορεί να καθοριστεί ως εξής:

$$D_j(x, y) = \begin{cases} 0 & \text{if } I_j(x, y) = I_{j+1}(x, y) \\ 1 & \text{if } I_j(x, y) \neq I_{j+1}(x, y) \end{cases}$$

(2)

Όπου το  $j$  είναι ο αριθμός καρτέ της ακολουθίας εικόνων και  $(x, y)$  οι τιμές συντεταγμένων στη δισδιάστατη εικόνα.

#### 2.2.4 Κατασκευή της Μέσης Διαφορικής Εικόνας Βαδίσματος

Με την επικάλυψη όλων των εικόνων μαζί ενός πλήρους κύκλου βαδίσματος, μπορούμε να πάρουμε την ακόλουθη μέση διαφορική εικόνα βαδίσματος.

$$G(x, y) = \frac{1}{N-1} \sum_{j=1}^{N-1} D_j(x, y)$$

(3)

όπου  $N$  είναι ο αριθμός των καρτέ σε έναν ολοκληρωμένο κύκλο βαδίσματος μιας ακολουθίας σιλουέτας. Η εικόνα 2.2.3.2 δείχνει τις διαφορικές εικόνες σε έναν κύκλο βαδίσματος και τη μέση διαφορική εικόνα βαδίσματος, αντίστοιχα.



Εικόνα 2.2.3. 2 Διαφορικές εικόνες και μέση διαφορική εικόνα βαδίσματος

## 2.3 Εξαγωγή Χαρακτηριστικών

Παρά το γεγονός ότι στην προηγούμενη ενότητα έχουμε συμπίεσει τα ανθρώπινα χαρακτηριστικά του βαδίσματος σε μια εικόνα, η διάσταση της μέσης διαφορικής εικόνας βαδίσματος εξακολουθεί να είναι πολύ μεγάλη. Η πιο συχνά χρησιμοποιούμενη μέθοδος μείωσης διαστάσεων είναι η ανάλυση κυρίων συνιστωσών (PCA). Στην παραδοσιακή μέθοδο PCA, κάθε δισδιάστατη εικόνα πρέπει να μετατραπεί σε ένα μονοδιάστατο διάνυσμα, οδηγώντας σε ένα πίνακα συν-διακύμανσης (covariance matrix) με μεγάλο μέγεθος. Αυτός ο μεγάλος πίνακας θα χρησιμοποιήσει υπερβολικά μεγάλη μνήμη αποθήκευσης και είναι δύσκολο να αξιολογηθεί με ακρίβεια.

Για να μειωθεί το μέγεθος της μνήμης που χρειάζεται για την αποθήκευση των δεδομένων και να επιταχυνθούν οι υπολογισμοί, υιοθετήθηκε η δισδιάστατη ανάλυση κυρίων συνιστωσών (2DPCA) με την οποία γίνεται μείωση του πλήθους των διαστάσεων. Αυτή προτάθηκε για πρώτη φορά από τους Yang και λοιπούς [27] στην ανακατασκευή ανθρώπινου προσώπου. Ο τελικός μας στόχος είναι η προβολή μιας μέσης διαφορικής εικόνας βαδίσματος  $G$ , ενός  $m * n$  πίνακα, σε ένα διάνυσμα  $m$  διαστάσεων  $Y$ , το οποίο ονομάζεται προβαλλόμενο διάνυσμα χαρακτηριστικών της εικόνας  $G$  από τον ακόλουθο γραμμικό μετασχηματισμό:

$$Y = GW$$

(4)

Όπου  $W$  δηλώνει ένα μοναδιαίο διάνυσμα στήλης  $n$  στοιχείων. Για να διατηρήσουμε τα χαρακτηριστικά του  $G$ , το  $W$  πρέπει να κάνει το  $Y$  να έχει τη μέγιστη διασπορά. Ορίζουμε  $S_y$  ως τη διασπορά του  $Y$  ως εξής:

$$\begin{aligned} S_y &= E (Y - E (Y)) (Y - E (Y))^T \\ &= E (GW - E (GW)) (GW - E (GW))^T \\ &= E (G - E (G)) W W^T (G - E (G))^T \end{aligned}$$

(5)

Το ίχνος (trace) του  $S_y$  μπορεί να εκφραστεί ως:

$$\text{tr}(S_y) = W^T \left( E(G - E(G))^T (G - E(G)) \right) W$$

(6)

Εδώ, μπορούμε να ορίσουμε τον πίνακα συνδιακύμανσης εικόνας ως:

$$C_t = E(G - E(G))^T (G - E(G))$$

(7)

Στην εργασία αυτή, η μέση διαφορική εικόνα βαδίσματος για κάθε ένα άτομο ξεχωριστά (1,2...M) εκφράζεται ως  $G_1, G_2 \dots G_M$  και στην συνέχεια το  $C_t$  μπορεί να υπολογιστεί με:

$$C_t = \sum_{i=1}^M (G_i - \bar{G})^T (G_i - \bar{G})$$

(8)

Στόχος μας είναι να βρούμε μια σειρά από  $W_{\text{opt}}$  στην εξίσωση (6) ώστε το  $\text{tr}(S_y)$  να έχει τη μέγιστη τιμή. Σύμφωνα με το [13], ο βέλτιστος άξονας προβολής  $W_{\text{opt}}$  είναι το μοναδιαίο ορθογώνιο ιδιοδιάνυσμα του  $C_t$  που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή. Ορίζουμε τα πρώτα  $d$  μοναδιαία ορθογώνια ιδιοδιανύσματα  $W_1, W_2, \dots, W_d$ , ως εξής:

$$\{W_1, \dots, W_d\} = \arg \max (W^T C_t W)$$

$$W_i W_j = 0, \quad i \neq j, \quad i, j = 1, \dots, d$$

$$(9) \quad W_i W_j = 1, \quad i = j, \quad j = 1, \dots, d$$

Τα πρώτα  $d$  βέλτιστα διανύσματα προβολής,  $W_1, W_2, \dots, W_d$ , χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών από τις μέσες διαφορικές εικόνες. Δηλαδή, έχοντας μια μέση διαφορική εικόνα βαδίσματος  $X$ , θεωρούμε

$$(10) \quad Y_k = G W_k \quad k = 1, 2, \dots, d$$

Στην συνέχεια παίρνουμε μια σειρά των προβληθέντων χαρακτηριστικών των διανυσμάτων  $Y_1, Y_2, \dots, Y_d$ , τα οποία είναι διαφορετικά από τα βαθμωτά ισοδύναμα που λαμβάνονται από την PCA. Με τη χρήση 2DPCA η αρχική  $m * n$  εικόνα προβάλλεται σε  $m * d$  ( $d \leq n$ ) πίνακα χαρακτηριστικών  $Y$  ως

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_d \end{bmatrix}$$

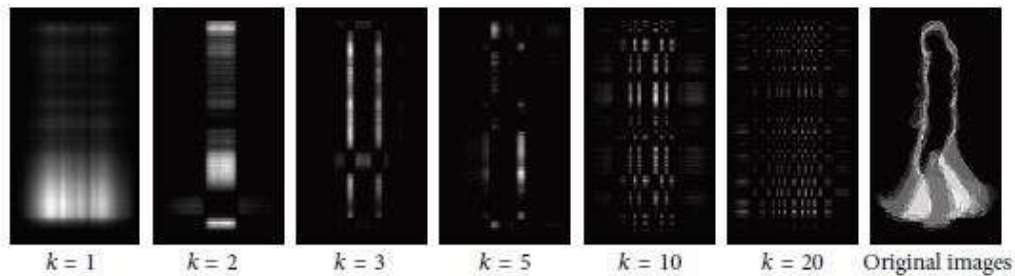
(11)

Η απόσταση μεταξύ δύο πινάκων χαρακτηριστικών ορίζεται ως:

$$d(Y(i), Y(j)) = \sum_{k=1}^d \left\| Y_k^{(i)} - Y_k^{(j)} \right\|,$$

(12)

Όπου  $\|Y_k^{(i)} - Y_k^{(j)}\|$  σημαίνει την Ευκλείδεια απόσταση μεταξύ των δύο φορέων



Εικόνα 2.3. 1 Οι ανακατασκευασμένες υπο-εικόνες ( $k= 1,2,3,5,10,20$ ) και οι αρχικές εικόνες

## 2.4 Αναγνώριση Σιλουετών

Ακολουθώντας το μοτίβο που πρότειναν οι Sarkar και λοιποί [22], αξιολογούμε τις επιδόσεις για το σενάριο της αναγνώρισης.

Στο σενάριο της αναγνώρισης, κάθε ακολουθία εικόνων στην gallery (σιλουέτες εκπαίδευσης) μετατρέπεται σε έναν  $m * d$  πίνακα χαρακτηριστικών  $Y(i)$ , σύμφωνα με την μέθοδο που περιγράψαμε στην Παράγραφο 2.3. Έχοντας μια ακολουθία probe (σιλουέτες εξέτασης), αυτή μετατρέπεται σε πίνακα χαρακτηριστικών που ορίζεται ως  $P$ . Αυτός ο πίνακας χαρακτηριστικών  $P$  αντιστοιχείται στο άτομο  $k$  χρησιμοποιώντας τη μέθοδο του πλησιέστερου γείτονα:

$$d(Y(k), P) = \min_i d(Y(i) - P)$$

(13)

## 2.5 Ανακατασκευή με βάση τη μέση διαφορική εικόνα βαδίσματος

Στην PCA οι κύριες συνιστώσες και τα ιδιοδιανύσματα μπορούν να συνδυαστούν ώστε να ανακατασκευαστεί ο αρχικός πίνακας. Ομοίως, η 2DPCA μπορεί επίσης να χρησιμοποιηθεί για την ανακατασκευή της μέσης διαφορικής σιλουέτας βαδίσματος.

Ας υποθέσουμε ότι τα ιδιοδιανύσματα που αντιστοιχούν στις μεγαλύτερες  $d$  τιμές ιδιοδιανυσμάτων του  $C_1$  είναι  $W_1 \dots W_d$ . Αυτό θα μας δώσει:

$$Y_k = GW_k \quad k = 1, \dots, d$$

$$[Y_1 \dots Y_d] = X [W_1 \dots W_d]$$

(16)

Σύμφωνα με τη συνάρτηση (9), τα  $W_1 \dots W_d$  είναι κανονικά ορθογώνια διανύσματα ώστε η νέα ανακατασκευασμένη  $G$  εικόνα να μπορεί να εκφραστεί ως:

$$\tilde{G} = G [W_1 \dots W_d] [W_1 \dots W_d]^T = [Y_1 \dots Y_d] [W_1 \dots W_d]^T$$

$$= \sum_{k=1}^d Y_k W_k^T.$$

(17)

## ΚΕΦΑΛΑΙΟ 3

### Περιβάλλον Ανάπτυξης Λογισμικού Matlab [30]

Το Matlab δημιουργήθηκε προς το τέλος της δεκαετίας του '70 από τον Cleve Moler, πρόεδρο του τμήματος πληροφορικής του πανεπιστημίου του Νέου Μεξικού. Ο σκοπός του σχεδιασμού του ήταν η πρόσβαση των σπουδαστών στις βιβλιοθήκες Linpack και Eispack χωρίς να έχουν γνώσεις Fortran. Σύντομα διαδόθηκε σε άλλα πανεπιστήμια και βρήκε ισχυρό ακροατήριο εντός της κοινότητας των εφαρμοσμένων μαθηματικών. Ο Jack Little, ένας μηχανικός, του οποίου το ενδιαφέρον κεντρίστηκε από μια επίσκεψη του Moler στο πανεπιστήμιο του Στάνφορντ το 1983, και αναγνωρίζοντας την εμπορική του ισχύ, συνεργάστηκε με τους Moler και Steve Bangert. Ξαναέγραψαν το Matlab σε γλώσσα C και ίδρυσαν την κοινότητα του Mathworks το 1984 για να συνεχίσουν την ανάπτυξή του. Οι τροποποιημένες βιβλιοθήκες του Matlab είναι γνωστές ως Jackpack. Το 2000, το Matlab ξαναγράφηκε για να χρησιμοποιήσει ένα σύνολο νεότερων βιβλιοθηκών για το χειρισμό πινάκων (Lapack).

#### 3.1 Τι είναι το Matlab;

Το Matlab (matrix laboratory) είναι ένα περιβάλλον αριθμητικής υπολογιστικότητας και προγραμματιστικής γλώσσας τέταρτης γενιάς. Το διαδραστικό περιβάλλον του προορίζεται για ανάπτυξη αλγόριθμων, επεξεργασία, ανάλυση και απεικόνιση δεδομένων, αριθμητικές μεθόδους επίλυσης προβλημάτων και μοντελοποίηση συστημάτων. Ξεκίνησε ως ένα πρόγραμμα «εργαστηρίου πινάκων» που είχε ως σκοπό να παρέχει αλληλεπιδρώσα προσπέλαση στις βιβλιοθήκες Linpack και Eispack. Από τότε έχει αναπτυχθεί αρκετά, για να γίνει ένα ισχυρότατο εργαλείο στην οπτικοποίηση, στον προγραμματισμό, στην έρευνα, στην επιστήμη των μηχανικών, και στις επικοινωνίες.

Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων, ωστόσο είναι πολύ “ισχυρό” και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές



από την C++ όπως την while, την switch και την if. Στον τομέα των γραφικών όσον αφορά τον μαθηματικό κλάδο μπορεί να υλοποιήσει συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών και άλλες. Όσον αφορά τον στατιστικό κλάδο μπορεί να υλοποιήσει ιστογράμματα, τομεογράμματα, ραβδογράμματα, εμβαδογράμματα και άλλα.

### 3.2 Το γραφικό περιβάλλον χρήστη – GUI

Στο προγραμματιστικό μέρος της εργασίας έγινε χρήση του γραφικού περιβάλλοντος χρήστη GUI (Graphical User Interface) του Matlab. Γενικά, ως γραφικό περιβάλλον εννοούμε ένα σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη μιας ψηφιακής συσκευής (π.χ. Η/Υ) με σκοπό την αλληλεπίδραση μεταξύ του χρήστη και της συσκευής. Παρέχουν στον τελευταίο, μέσω γραφικών, ενδείξεις και εργαλεία προκειμένου αυτός να φέρει εις πέρας κάποιες επιθυμητές λειτουργίες. Για τον λόγο αυτό δέχονται και είσοδο από τον χρήστη και αντιδρούν ανάλογα στα συμβάντα που αυτός προκαλεί με τη βοήθεια κάποιας συσκευής εισόδου (π.χ. πληκτρολόγιο, ποντίκι).

Τα περισσότερα σύγχρονα προγράμματα και λειτουργικά συστήματα υπολογιστών, προσφέρουν στους χρήστες τους κάποιο GUI γιατί αυτός ο τρόπος αλληλεπίδρασης με τον υπολογιστή ταιριάζει αρκετά στην ανθρώπινη εμπειρία και φύση. Σωστά σχεδιασμένα γραφικά προσφέρουν ένα όμορφο, εύχρηστο και λειτουργικό περιβάλλον εργασίας. Πριν από την έλευση και καθιέρωση των GUI ο κανόνας στους μικροϋπολογιστές ήταν η αλληλεπίδραση με τον χρήστη μέσω κάποιου κέλφους γραμμής εντολών.

Συγκεκριμένα, ο κύριος λόγος χρήσης του GUI στο Matlab είναι η ευκολία που παρέχει στους χρήστες του προγράμματος. Εάν δεν υπήρχε το GUI, οι χρήστες θα έπρεπε να εργαστούν από τη διεπαφή γραμμής εντολών, πράγμα εξαιρετικά δύσκολο και πολύπλοκο για χρήστες οι οποίοι δεν είναι καλοί γνώστες προγραμματισμού. Φανταστείτε εάν έπρεπε να εισάγετε εντοές για την αναζήτησή σας σε μια μηχανή αναζήτησης Ιστού, αφού ακόμα και η αναζήτηση Ιστού είναι ένα GUI επίσης.

## ΚΕΦΑΛΑΙΟ 4

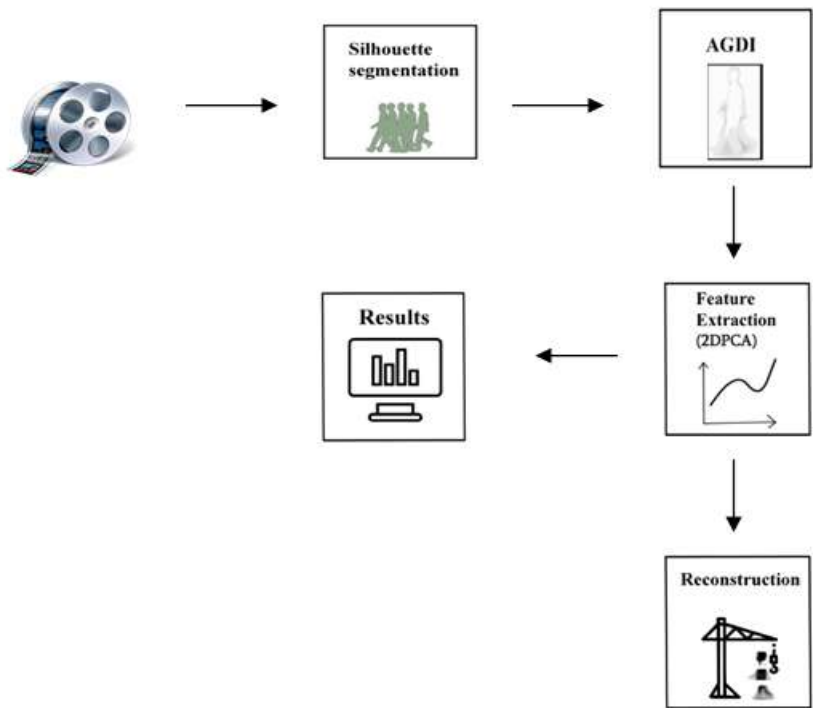
### Υλοποίηση Αναγνώρισης Ανθρώπινου Βαδίσματος

#### 4.1 Γενικά

Για την υλοποίηση του αλγορίθμου χρησιμοποιούμε τις δυνατότητες που μας δίνει το προγραμματιστικό περιβάλλον της Matlab. Η υπάρχουσα βάση δεδομένων είναι η Casia Gait Database (Database B) [29,30], μια από τις μεγαλύτερες βάσεις δεδομένων στην έρευνα του ανθρώπινου βαδίσματος.

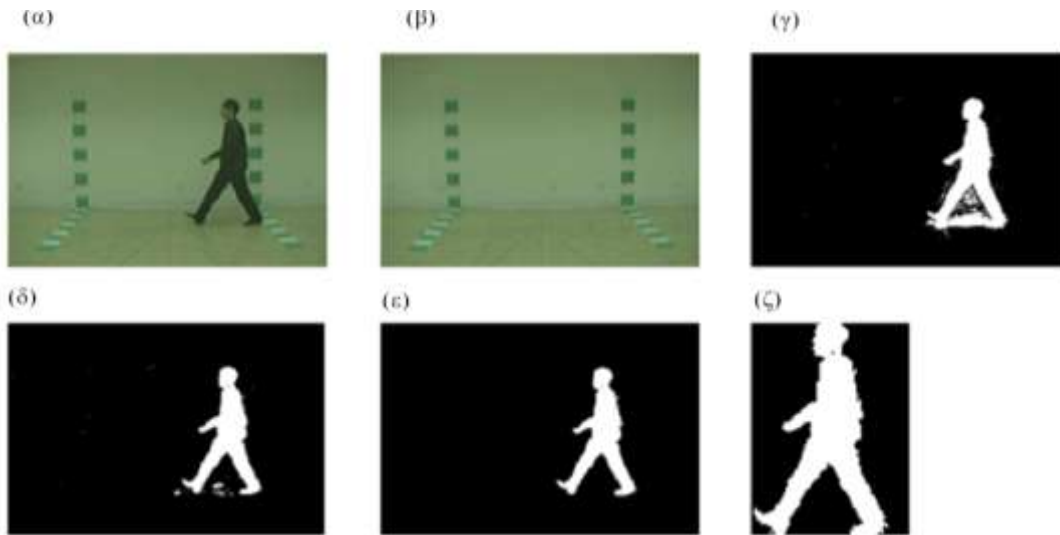
Μέσα στη βάση δεδομένων περιέχονται 86 αντικείμενα (άτομα), 66 άντρες και 20 γυναίκες. Κάθε άτομο είναι τραβηγμένο από 11 διαφορετικές γωνίες (από 0° έως 180°, με διάστημα 18°). Για κάθε άτομο υπάρχουν 6 normal walking ακολουθίες (nm-01 έως nm-06), 2 cloak ακολουθίες (cl-01 και cl-02) και 2 bag ακολουθίες (bg-01 και bg-02) που έχουν ληφθεί η κάθε μια και από 11 γωνίες. Κάθε ακολουθία περπατήματος περιέχει 3-8 κύκλους βάδισης (40-100 καρέ). Το μέγεθος του κάθε καρέ είναι 320 x 240 εικονοστοιχεία.

Για το πείραμά μας χρησιμοποιούμε όλα τα άτομα από τις δύο πρώτες normal walking. Για training data (gallery) επιλέγουμε από τη normal-01, όλα τα καρέ που εμπεριέχονται από όλες τις γωνίες. Για test data (probe) επιλέγουμε από τη normal-02, όλα τα καρέ από όλες τις γωνίες. Στη μέθοδο της εξαγωγής χαρακτηριστικών, 2DPCA, επιλέγουμε τις 20 μεγαλύτερες ιδιοτιμές. Για την αναγνώριση χρησιμοποιούμε την ευκλείδεια απόσταση, επιλέγοντας πάντα τη μικρότερη απόσταση. Σαν τελευταίο επιπλέον βήμα έχουμε την ανακατασκευή την εικόνας μέσω των χαρακτηριστικών της.



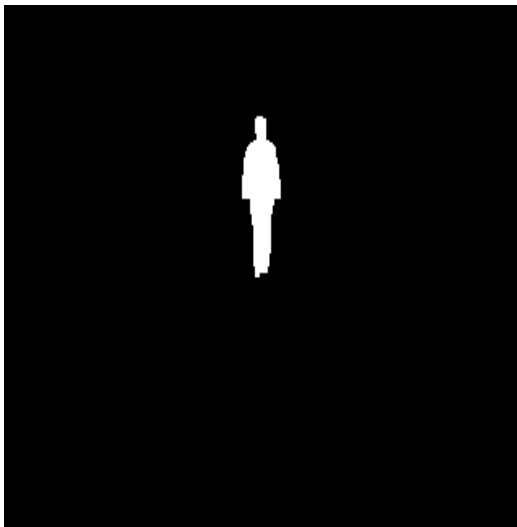
Εικόνα 4.1. 1 Σχεδιάγραμμα Αναγνώρισης

Στην παρούσα πτυχιακή εργασία δουλεύουμε με έτοιμη βάση δεδομένων ασπρόμαυρων εικόνων. Όμως, αν δεν είχαμε την ήδη υπάρχουσα βάση, θα μπορούσαμε να δημιουργήσουμε τις κατάλληλες ασπρόμαυρες εικόνες. Για τη διαδικασία αυτή χρησιμοποιούμε το μοντέλο και φίλτρο Gauss, όπως φαίνεται στην παρακάτω εικόνα.

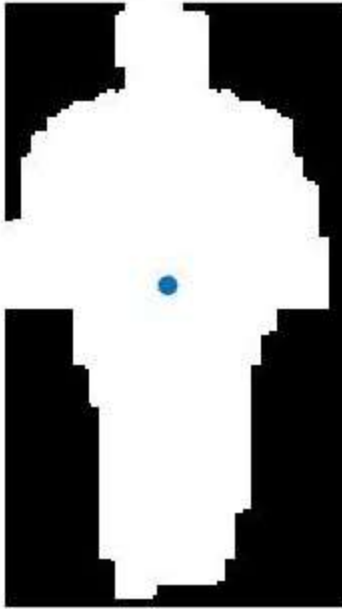


Εικόνα 4.1. 2 (α) Ακατέργαστη εικόνα , (β) Φόντο , (γ) Αφαίρεση Φόντου ,  
 (δ) Εξάλειψη Σκιάς , (ε) Μετά την Επεξεργασία , (ζ) Κανονικοποιημένη  
 Σιλουέττα

Όπως προαναφέραμε, εφόσον έχουμε ήδη ασπρόμαυρες εικόνες, προχωράμε στο επόμενο βήμα το οποίο είναι η κανονικοποίηση και η εύρεση κέντρου ισορροπίας. Το αρχικό μέγεθος της εικόνας είναι 320x240, και μετά την επεξεργασία γίνεται 160x90. Για την κανονικοποίηση χρησιμοποιούμε τη μέθοδο “bicubic interpolation”.



Εικόνα 4.1. 3 Αρχική Εικόνα, 0 μόιρες



Εικόνα 4.1. 4 Κανονικοποιημένη σιλουεττα με το κέντρο της, 0 μοίρες

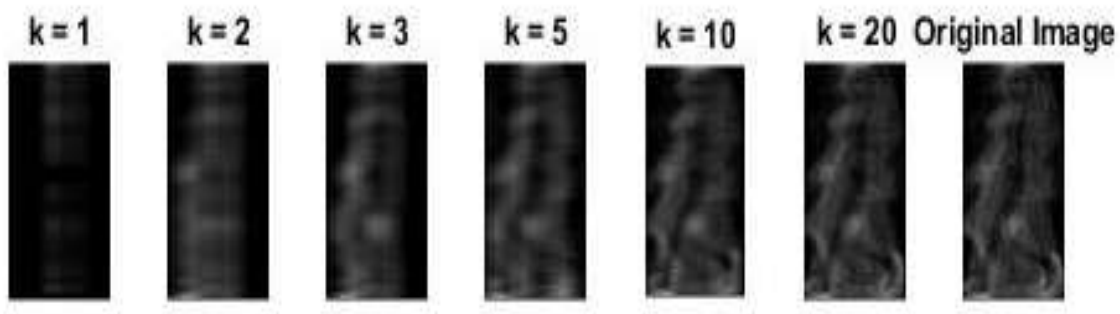
Σημαντικό ρόλο έχει η διαφορά των εικόνων, καθώς από αυτές τις διαφορές εμφανίζουμε και την μέση διαφορική εικόνα που επιζητούμε κάνοντας αυτή τη διαδικασία. Η διαφορά μεταξύ των εικόνων γίνεται εικονοστοιχείο με εικονοστοιχείο σύμφωνα με την θεωρία που έχουμε αναφέρει στο Κεφάλαιο 2. Η μέση διαφορική εικόνα δημιουργείται προσθέτοντας όλες τις εικόνες ενός πλήρους κύκλου βαδίσματος.



Εικόνα 4.1. 5 Διαφορικές εικόνες και μέση διαφορική εικόνα (AGDI), 90 μοίρες

Η εξαγωγή χαρακτηριστικών γίνεται με βάση το μοντέλο 2DPCA. Ο στόχος αυτής της τεχνικής είναι η εξοικονόμηση χώρου και χρόνου, καθώς και η δημιουργία μιας εικόνας από την οποία θα εξάγουμε τα χαρακτηριστικά. Χρησιμοποιούμε τις 20 μεγαλύτερες ιδιοτιμές δημιουργώντας μια «εικόνα χαρακτηριστικών». Η διαδικασία αυτή εφαρμόζεται και στις εικόνες gallery και στις probe. Έχοντας μια εικόνα χαρακτηριστικών από τις probe, την συγκρίνουμε με τις εικόνες του training. Η σύγκριση γίνεται με βάση την ευκλείδεια απόσταση, παίρνοντας πάντα τη μικρότερη απόσταση. Συγκρίνουμε ίδιες μοίρες και ψάχνουμε να βρούμε αν έχει αναγνωρίσει το άτομο, το οποίο του δώσαμε να συγκρίνει.

Έχοντας ολοκληρώσει την παραπάνω διαδικασία μπορούμε να ανακατασκευάσουμε την αρχική εικόνα σύμφωνα με τις ιδιοτιμές της καθεμίας, όπως φαίνεται στην Εικόνα 4.1.6 παρακάτω.

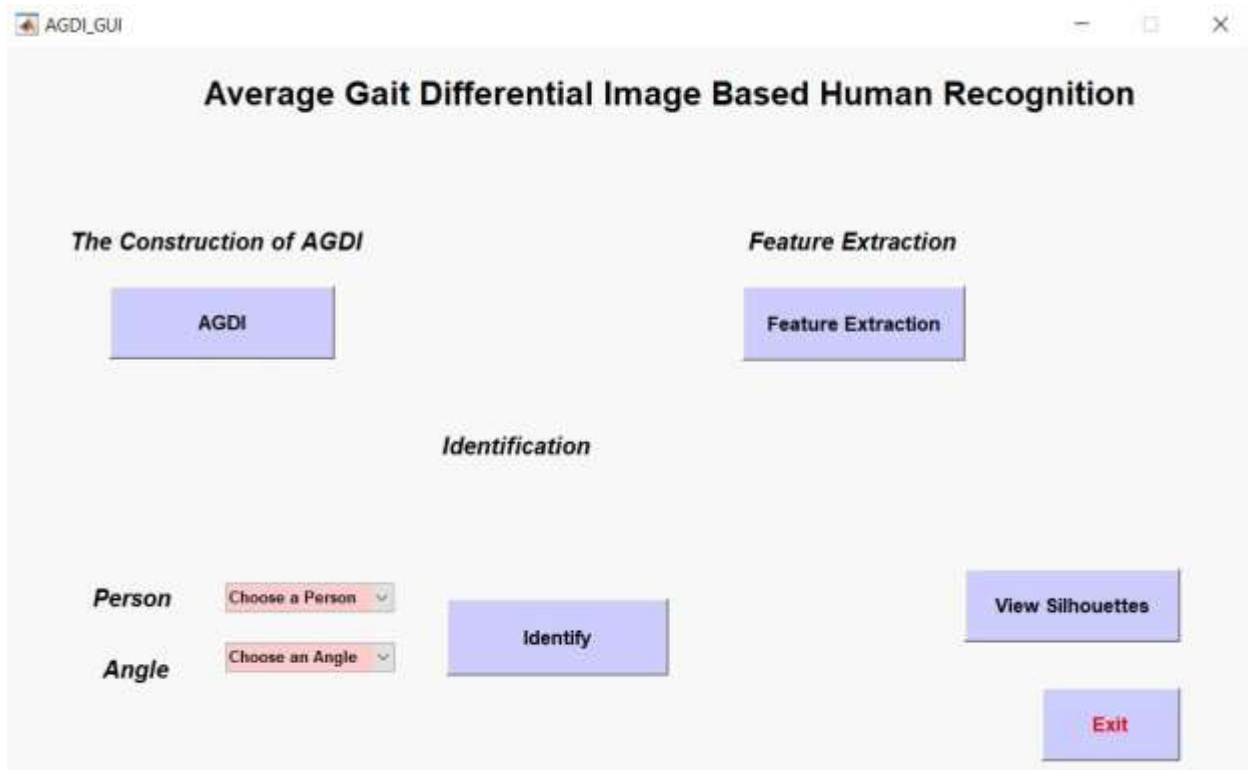


Εικόνα 4.1. 6 Ανακατασκευή Εικόνας Ατόμου 90 μοιρών και αρχική εικόνα.

## 4.2 Εφαρμογή Αναγνώρισης Ανθρώπινου Βαδίσματος

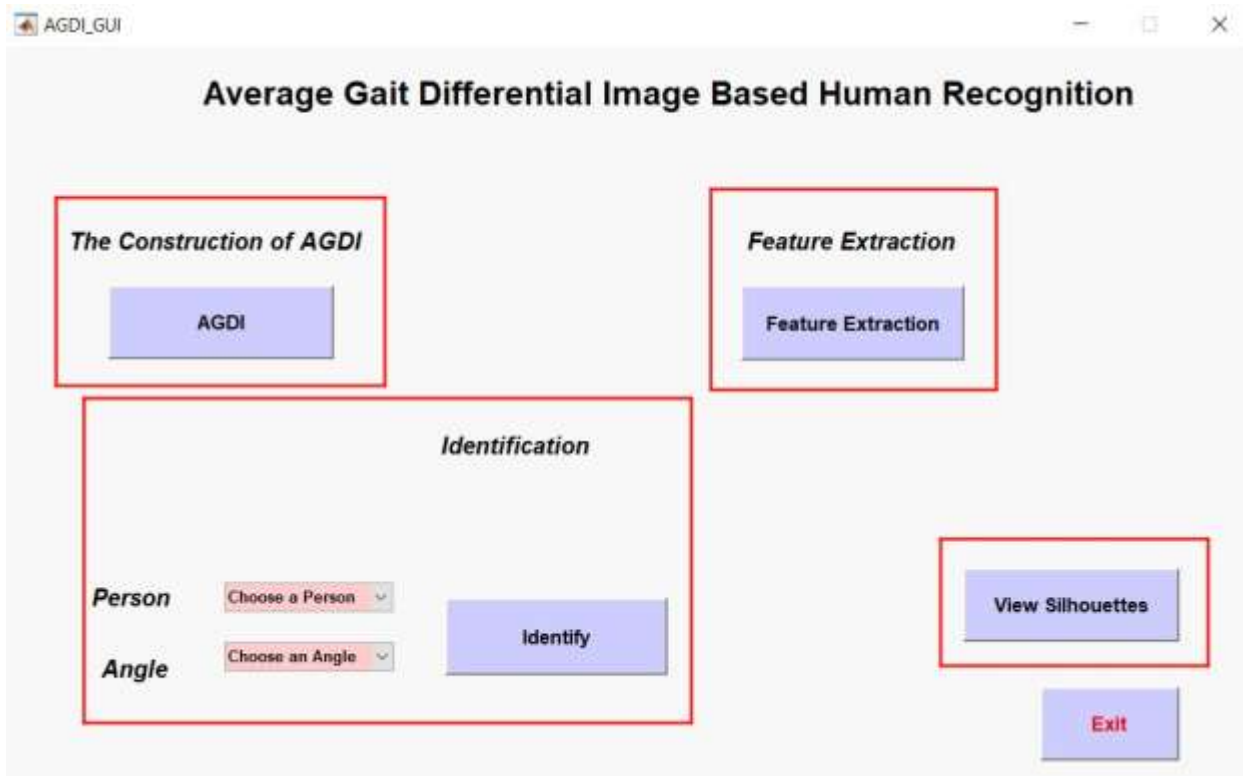
Παρακάτω παρουσιάζεται ένας αναλυτικός οδηγός της εφαρμογής που έχουμε δημιουργήσει για την αναγνώριση ανθρώπινου βαδίσματος σύμφωνα με τον αλγόριθμο που έχουμε αναλύσει στο Κεφάλαιο 2.

Αρχικά, ανοίγουμε το Matlab και επιλέγουμε τον φακέλο που περιέχει τις συναρτήσεις μας. Έπειτα, πηγαίνουμε στο command window και πληκτρολογούμε τη λέξη AGDI\_GUI. Αμέσως μας ανοίγει το γραφικό περιβάλλον και εμφανίζεται το παρακάτω παράθυρο:



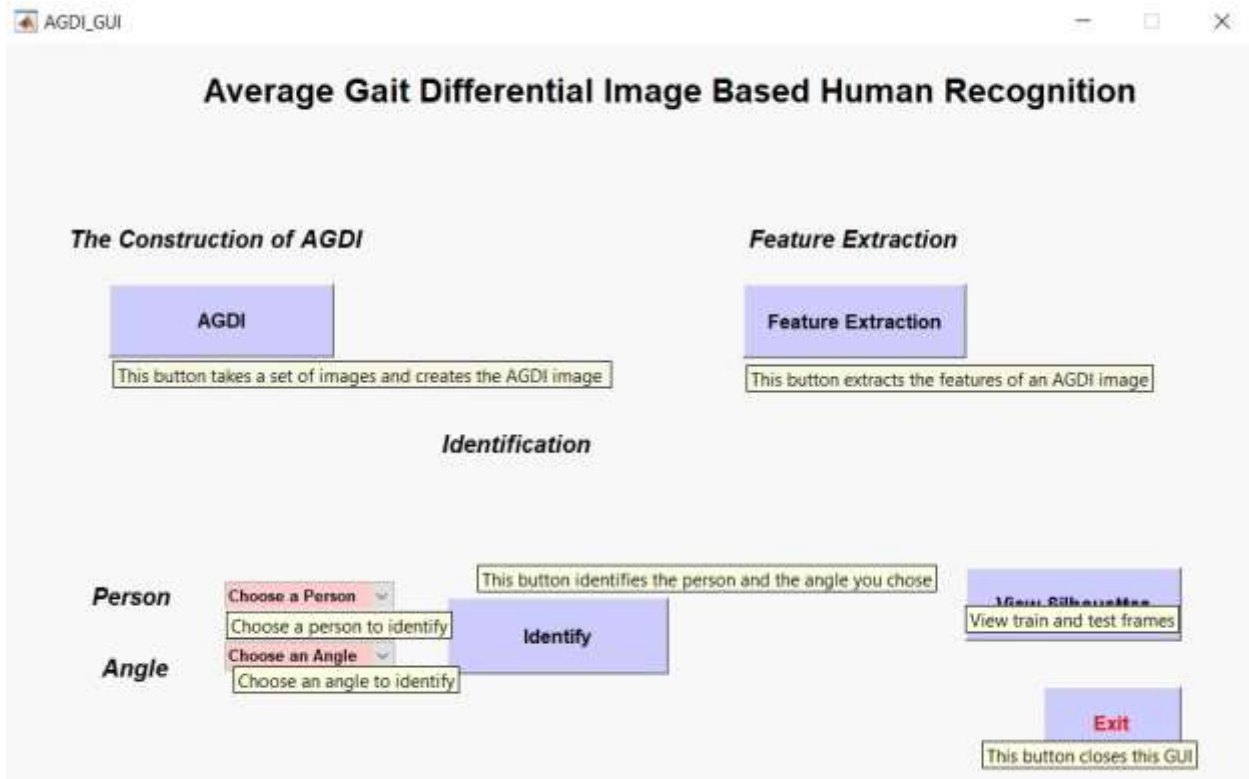
Εικόνα 4.2. 1 AGDI\_GUI, Γραφικό περιβάλλον

Η εφαρμογή μας είναι χωρισμένη σε τέσσερα βασικά μέρη τα οποία μπορούμε να διακρίνουμε στην παρακάτω εικόνα μέσα στα κόκκινα πλαίσια. Κάθε κουμπί έχει την αντίστοιχη ονομασία με συνάρτηση που τρέχει από πίσω. Επίσης, αν πάμε το δείκτη του ποντικιού μας πάνω από κάθε κουμπί θα εμφανιστεί μια περιγραφή για τη λειτουργία του κάθε κουμπιού. Τα τρία από τα τέσσερα κουμπιά πρέπει να τρέξουν με μια συγκεκριμένη σειρά ώστε η εφαρμογή να δουλέψει σωστά και να μας βγάλει τα επιθυμητά αποτελέσματα, δηλαδή την αναγνώριση ενός ατόμου. Το τέταρτο κουμπί ανοίγει ένα ξεχωριστό παράθυρο, στο οποίο μπορούμε να δούμε τα καρέ των εικόνων που θα χρησιμοποιήσουμε.



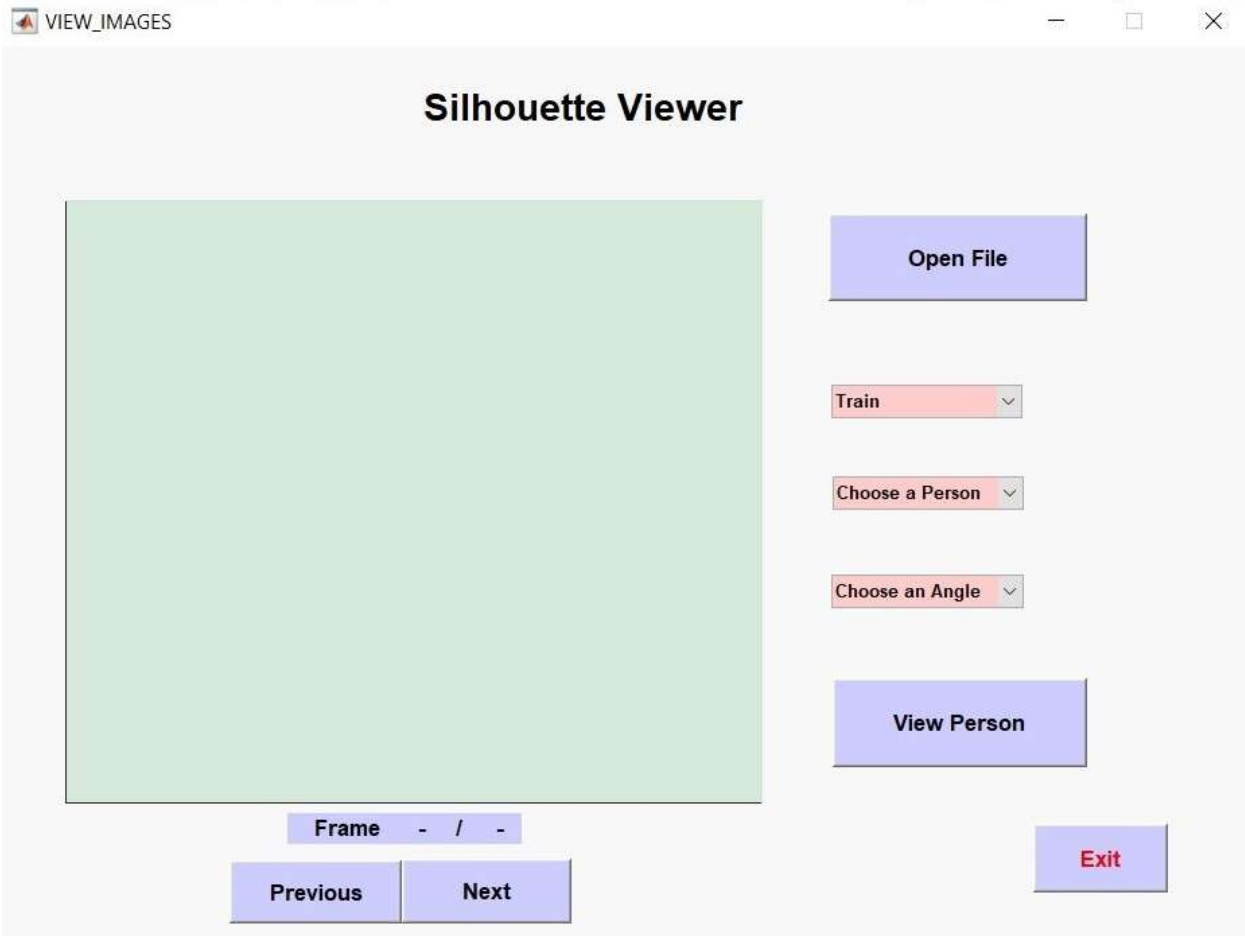
Εικόνα 4.2. 2 AGDI\_GUI, Βασικά Μέρη Εφαρμογής





Εικόνα 4.2. 3 AGDI\_GUI, Περιγραφές Buttons

Σαν πρώτο βήμα, αν θέλουμε να δούμε πρώτα τις εικόνες της βάσης δεδομένων μας, πατάμε το κουμπί «View Silhouettes» και μας εμφανίζεται ένα καινούριο παράθυρο με το όνομα «VIEW IMAGES».



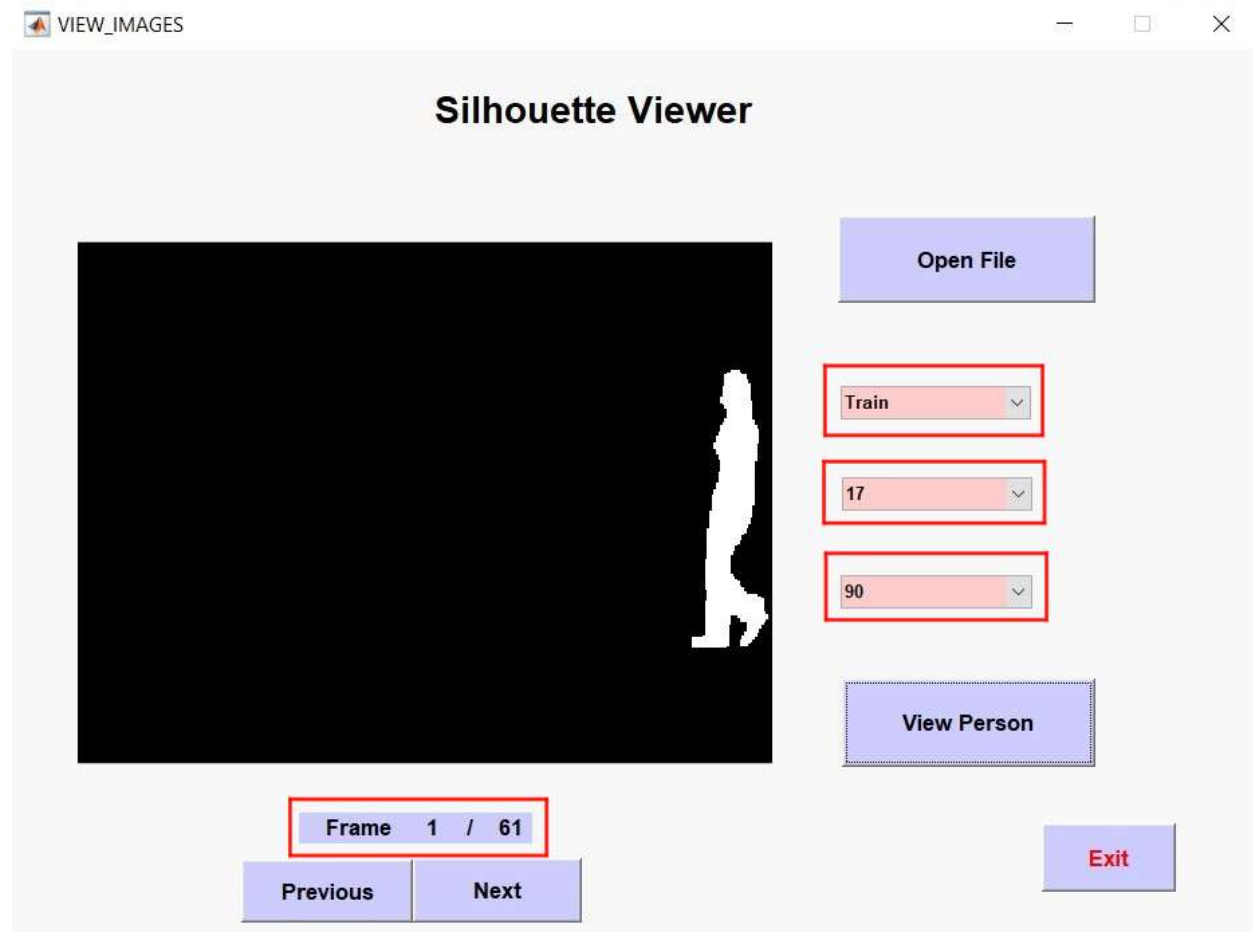
Εικόνα 4.2. 4 VIEW\_IMAGES, Silhouette Viewer

Σε αυτό το παράθυρο μπορούμε να δούμε τα καρέ των εικόνων τα οποία θα χρησιμοποιήσουμε για την αναγνώριση. Πατώντας το κουμπί «Open File» φορτώνουμε τις εικόνες που βρίσκονται στο φάκελο «silhouettes» .

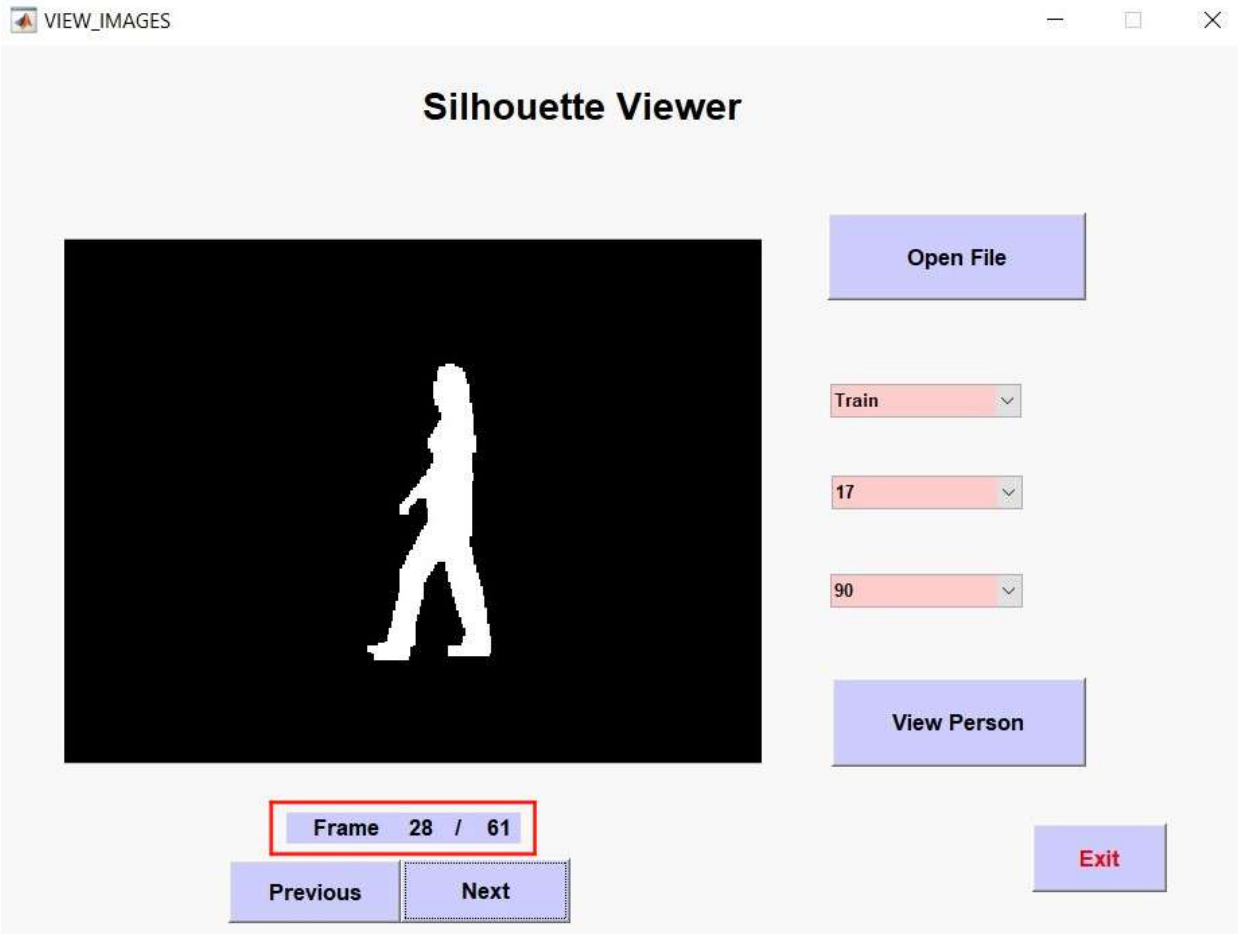
Όπως παρατηρούμε, υπάρχουν 3 list boxes. Στο 1<sup>ο</sup> επιλέγουμε αν θέλουμε να δούμε ένα άτομο από την κατηγορία Train ή Test. Στη συνέχεια, στο 2<sup>ο</sup> list box, επιλέγουμε ένα άτομο (σε αριθμό) που επιθυμούμε να δούμε και, τέλος, στο 3<sup>ο</sup> list box, τη γωνία που επιθυμούμε. Μόλις τελειώσουμε με τις επιλογές μας πατάμε το κουμπί «View Person». Αμέσως εμφανίζεται στα αριστερά το 1<sup>ο</sup> καρέ του ατόμου που επιλέξαμε και μπορούμε να δούμε όσα καρέ υπάρχουν μέσα στο φάκελό του, με τα κουμπιά «Previous» και «Next», που μας επιτρέπουν να πάμε στο

προηγούμενο ή επόμενο καρέ. Πάνω από αυτά τα κουμπιά βλέπουμε σε ποιο καρέ είμαστε αυτή τη στιγμή και πόσα καρέ έχει συνολικά το άτομο που έχουμε επιλέξει.

Παρακάτω παραθέτονται δύο εικόνες έχοντας επιλέξει κατηγορία «Train», άτομο «17» και μοίρα «90» στο πρώτο καρέ και σε κάποιο ενδιάμεσο καρέ.



Εικόνα 4.2. 5 Silhouette Viewer, Train/ Person 17/ Angle 90, Πρώτο Καρέ(61 συνολικά)

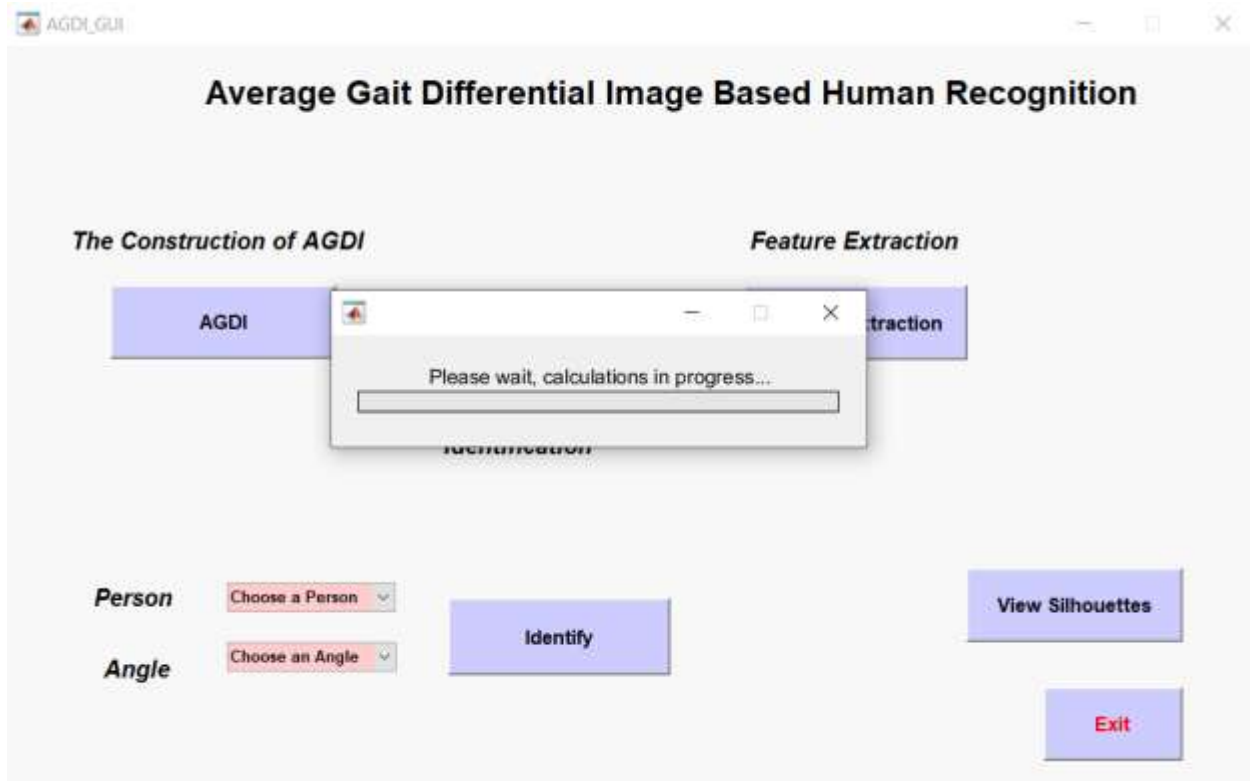


Εικόνα 4.2. 6 Silhouette Viewer, Train/ Person 17/ Angle 90, Καρέ 28

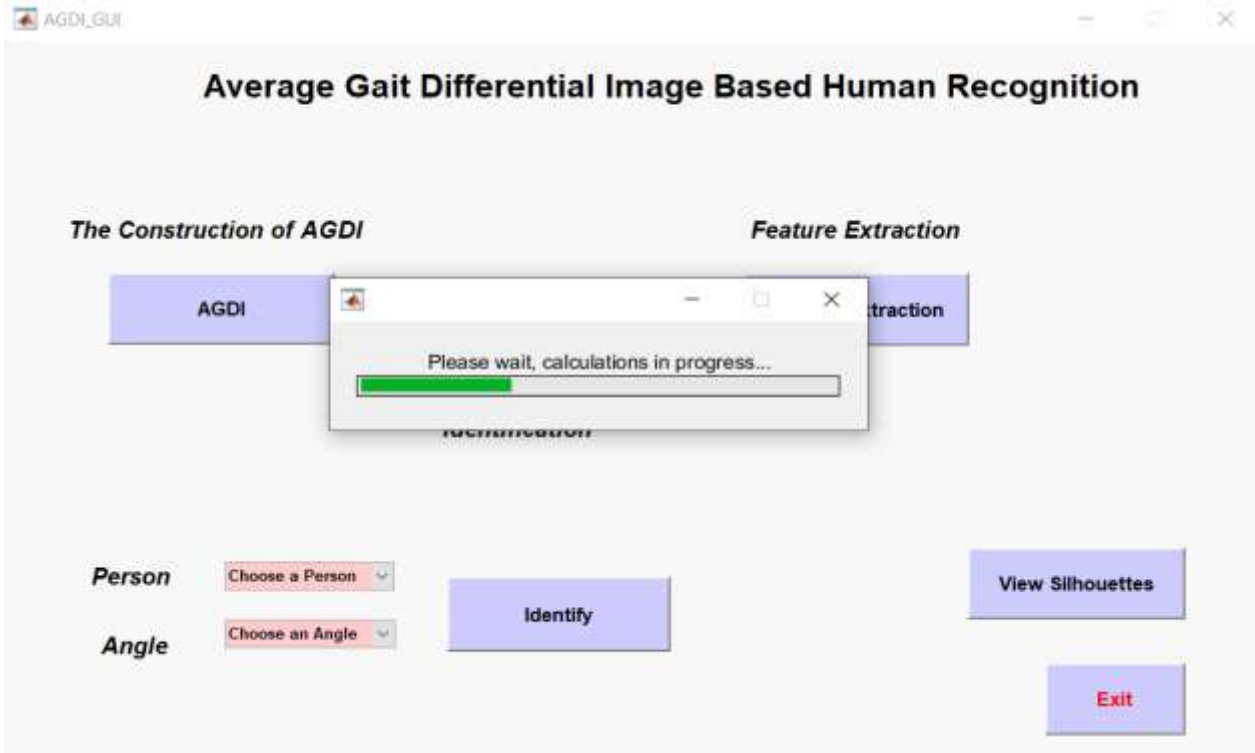
Για να δούμε κάποιο καρέ διαφορετικού ατόμου επιλέγουμε αυτό που επιθυμούμε και ξαναπατάμε το «View Person». Στο τέλος, για να κλείσουμε το παράθυρο πατάμε το «Exit» και επιστρέφουμε στην προηγούμενή μας φόρμα.

Όπως έχει αναφερθεί παραπάνω, τρία από τα τέσσερα βασικά κουμπιά της εφαρμογής πρέπει να λειτουργήσουν με κάποια συγκεκριμένη σειρά, ώστε να γίνει η αναγνώριση κάποιας φιγούρας. Ξεκινώντας πατάμε το κουμπί «AGDI», το οποίο κατασκευάζει την μέση εικόνα (AGDI). Μόλις το πατήσουμε, πρέπει να επιλέξουμε τον φάκελο που είναι αποθηκευμένες οι φιγούρες μας, «silhouettes» και αμέσως θα εμφανιστεί μια μπάρα προόδου. Περιμένουμε μέχρι να φτάσει στο τέλος και να γίνει το κουμπί «AGDI» πράσινο. Τότε έχει ολοκληρωθεί το πρώτο

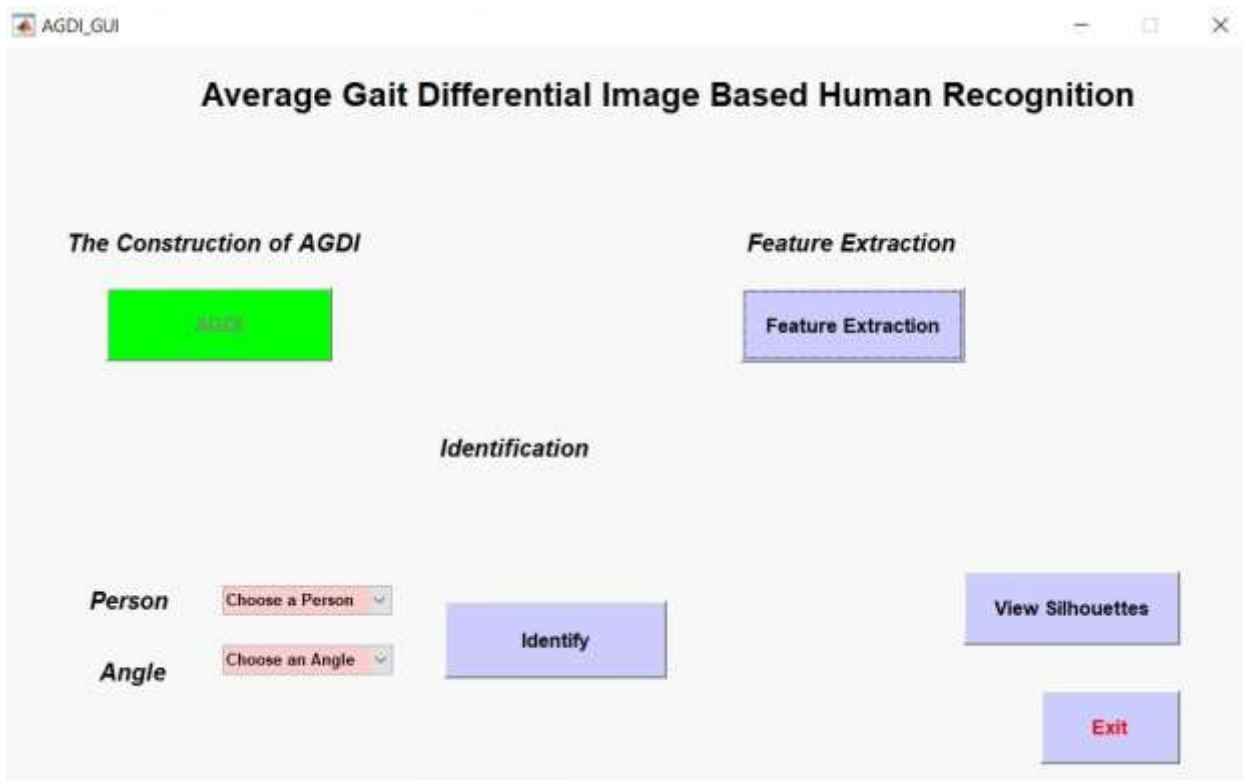
βήμα. Ο χρόνος διαδικασίας για αυτό το βήμα διαφέρει ανάλογα με τις επιδόσεις κάθε υπολογιστή, με έναν μέσο όρο 15 λεπτά(βλ. σελίδα 47).



Εικόνα 4.2. 7 Calculation of AGDI

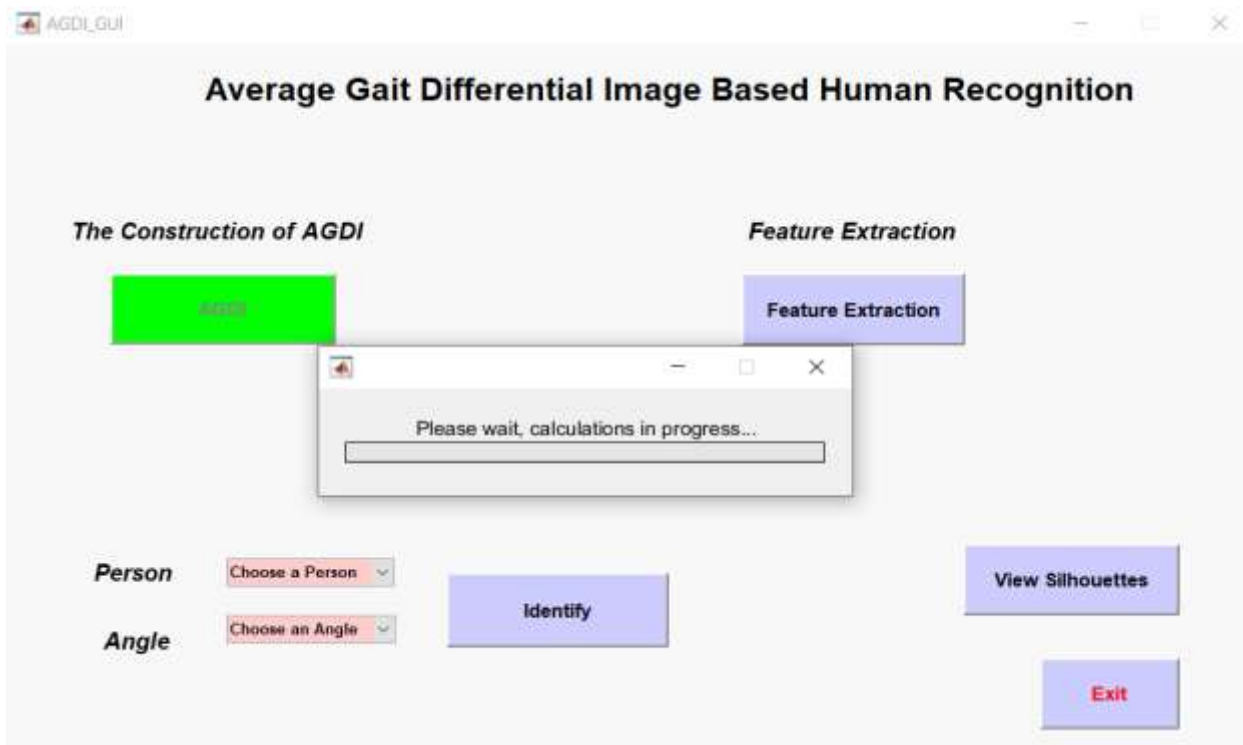


Εικόνα 4.2. 8 Calculation of AGDI, calculations in progress



Εικόνα 4.2. 9 Calculation of AGDI, Finished

Στη συνέχεια, προχωράμε στο δεύτερο βήμα το οποίο είναι η εξαγωγή των χαρακτηριστικών από τις εικόνες. Αυτή τη διαδικασία την κάνει το κουμπί «Feature Extraction». Οι υπολογισμοί εδώ είναι αρκετά πιο γρήγοροι, μερικά δευτερόλεπτα, και εμφανίζεται πάλι μια μπάρα πρόοδου και μόλις ολοκληρωθούν οι υπολογισμοί το κουμπί γίνεται πράσινο όπως και στο προηγούμενο.

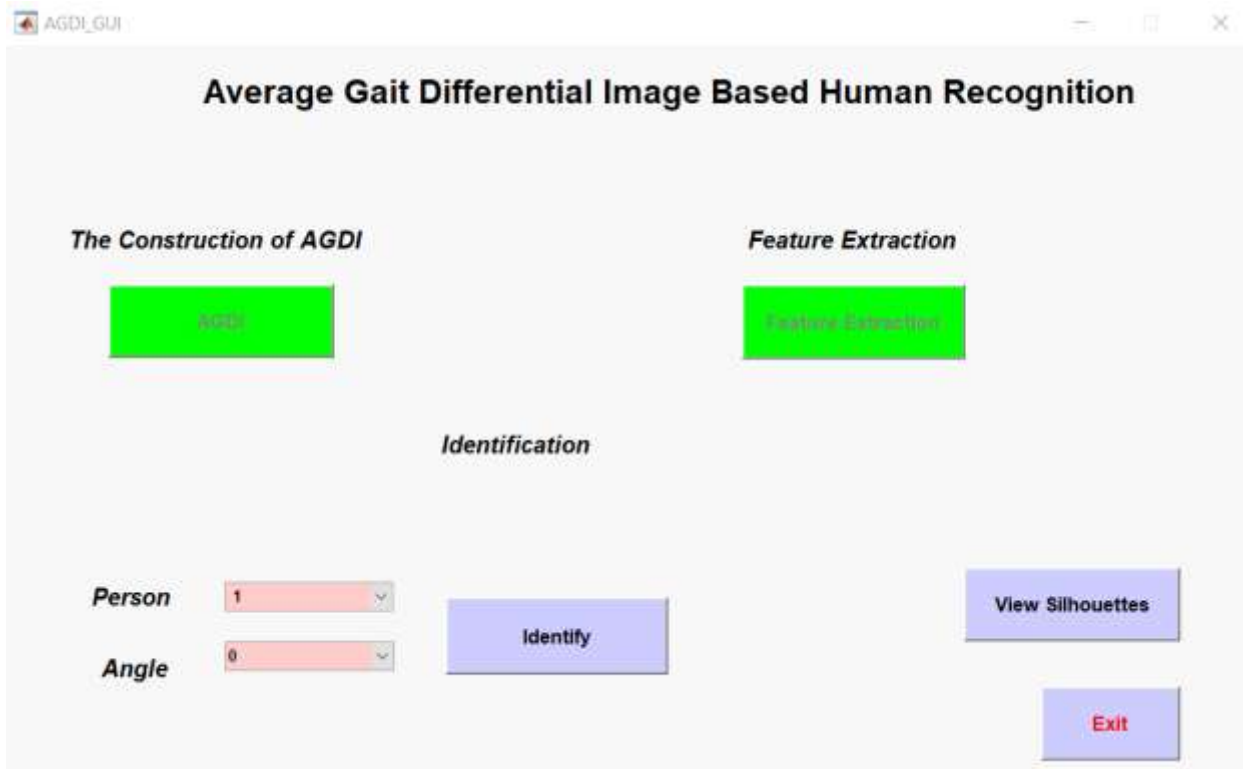


Εικόνα 4.2. 10 Feature Extraction



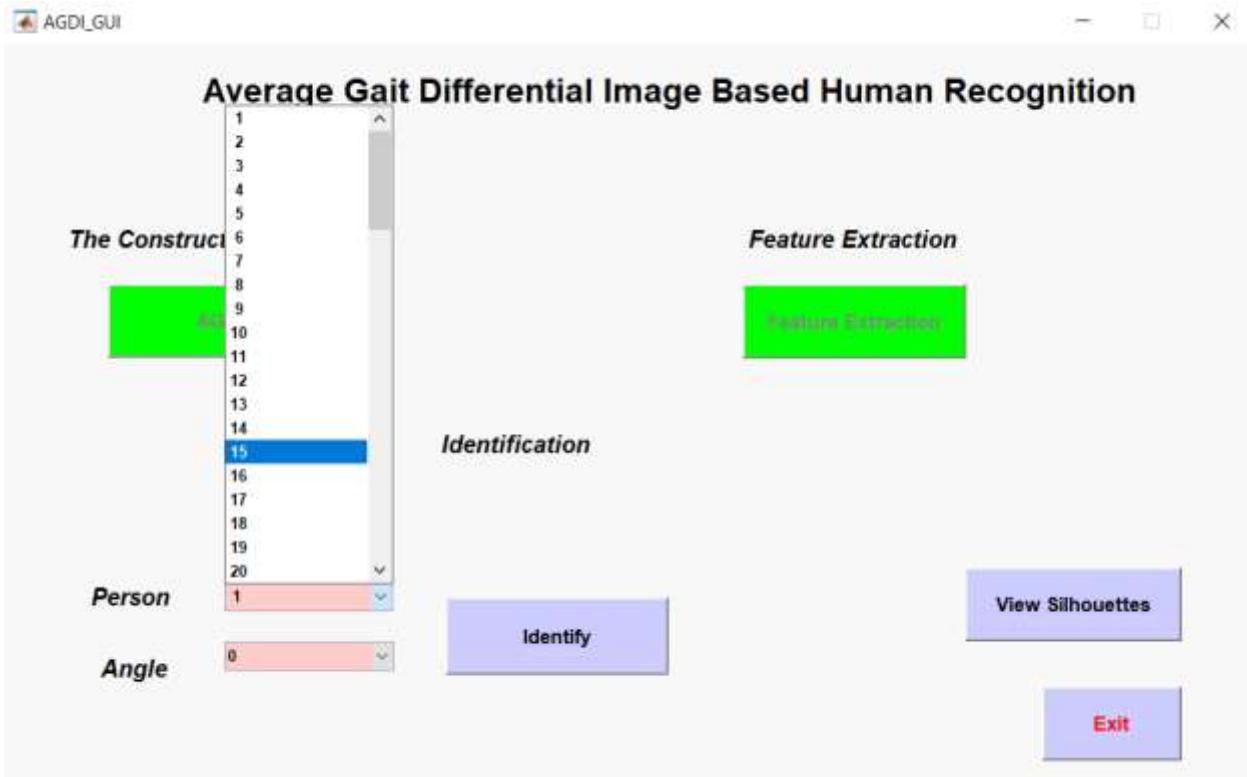
Εικόνα 4.2. 11 Feature Extraction, calculations in progress



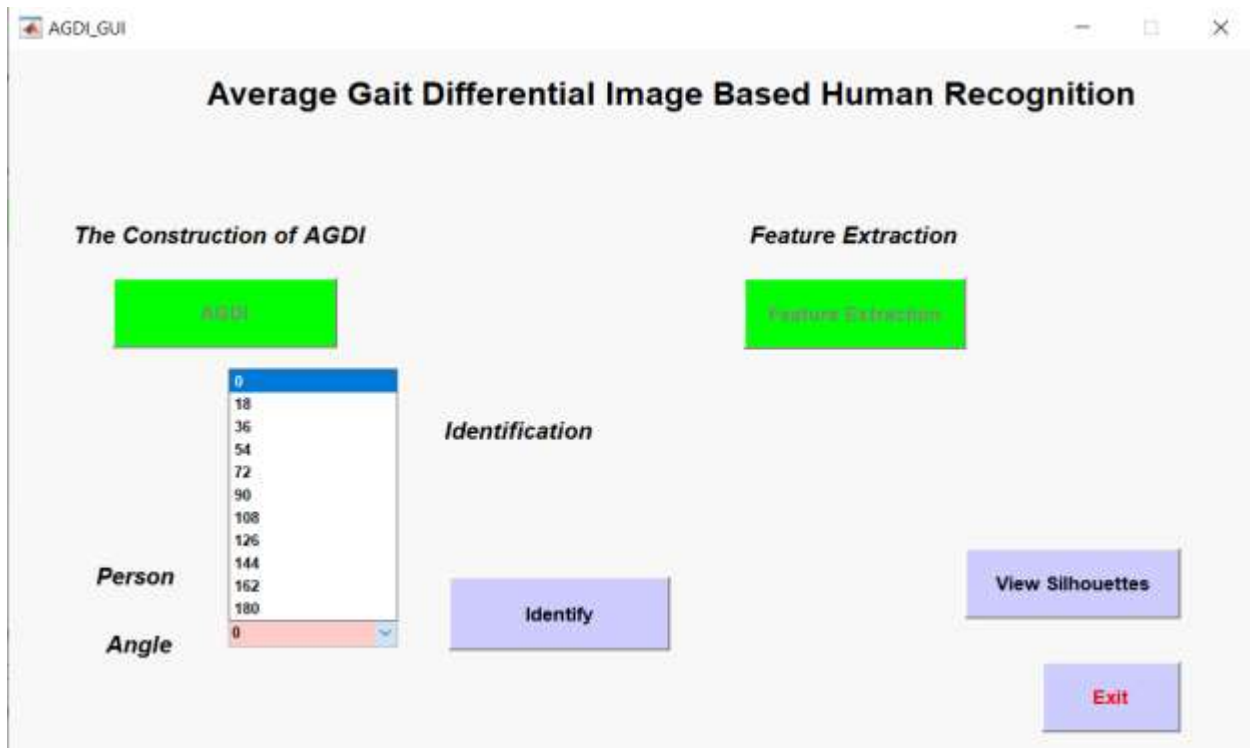


Εικόνα 4.2. 12 Feature Extraction, Finished

Ολοκληρώνοντας και το δεύτερο βήμα της εξαγωγής χαρακτηριστικών, προχωράμε στο τελευταίο και πιο σημαντικό, την αναγνώριση του ανθρώπινου βαδίσματος. Παρατηρούμε πως όταν γίνει πράσινο το κουμπί «Feature Extraction» εμφανίζονται οι επιλογές των δύο list boxes Person και Angle, όπου μπορούμε να επιλέξουμε το άτομο και τη γωνία που μας ενδιαφέρει να συγκρίνουμε.

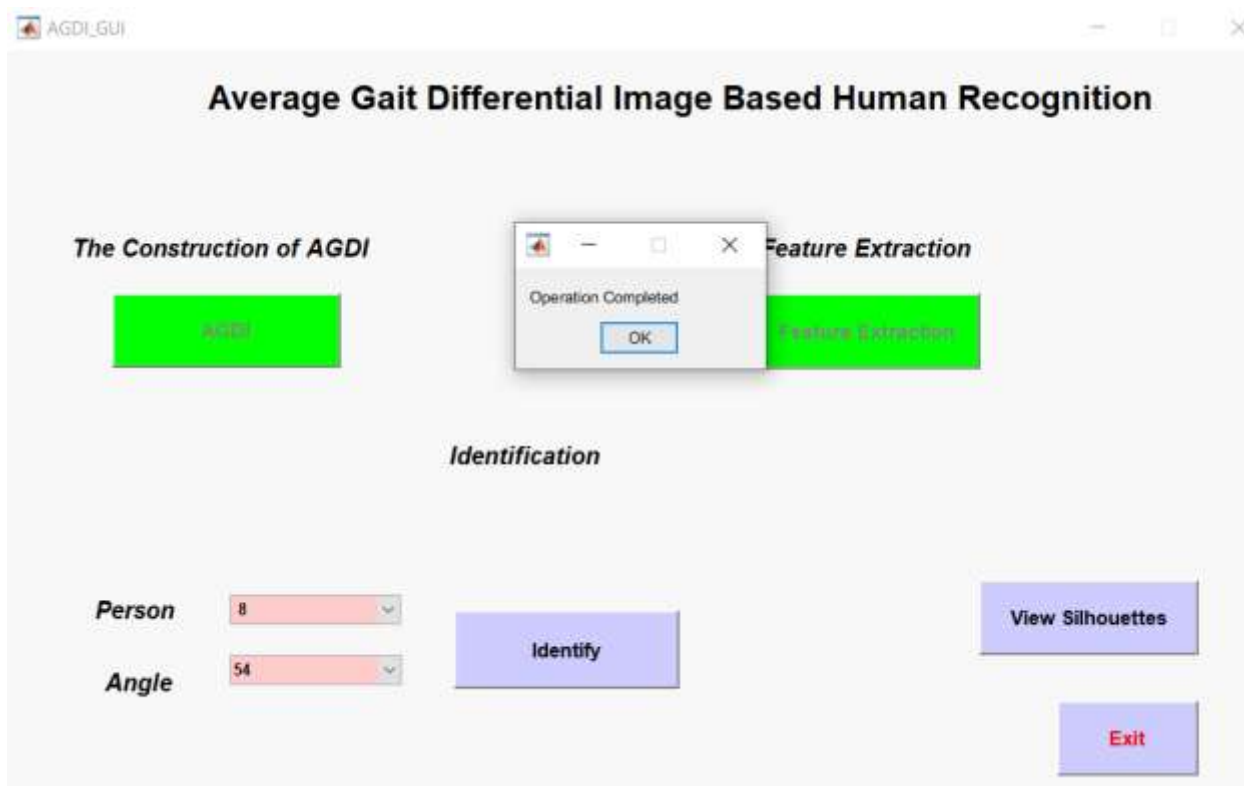


Εικόνα 4.2. 13 Choose a Person



Εικόνα 4.2. 14 Choose an Angle

Μόλις έχουμε επιλέξει αυτά που θέλουμε πατάμε το κουμπί «Identify». Θα εμφανιστεί για κλάσματα δευτερολέπτου ένα μήνυμα «Operation Completed» και αμέσως μετά ένα καινούριο παράθυρο, που θα μας ενημερώνει αν το άτομο που επιλέξαμε έχει αναγνωριστεί ή όχι.

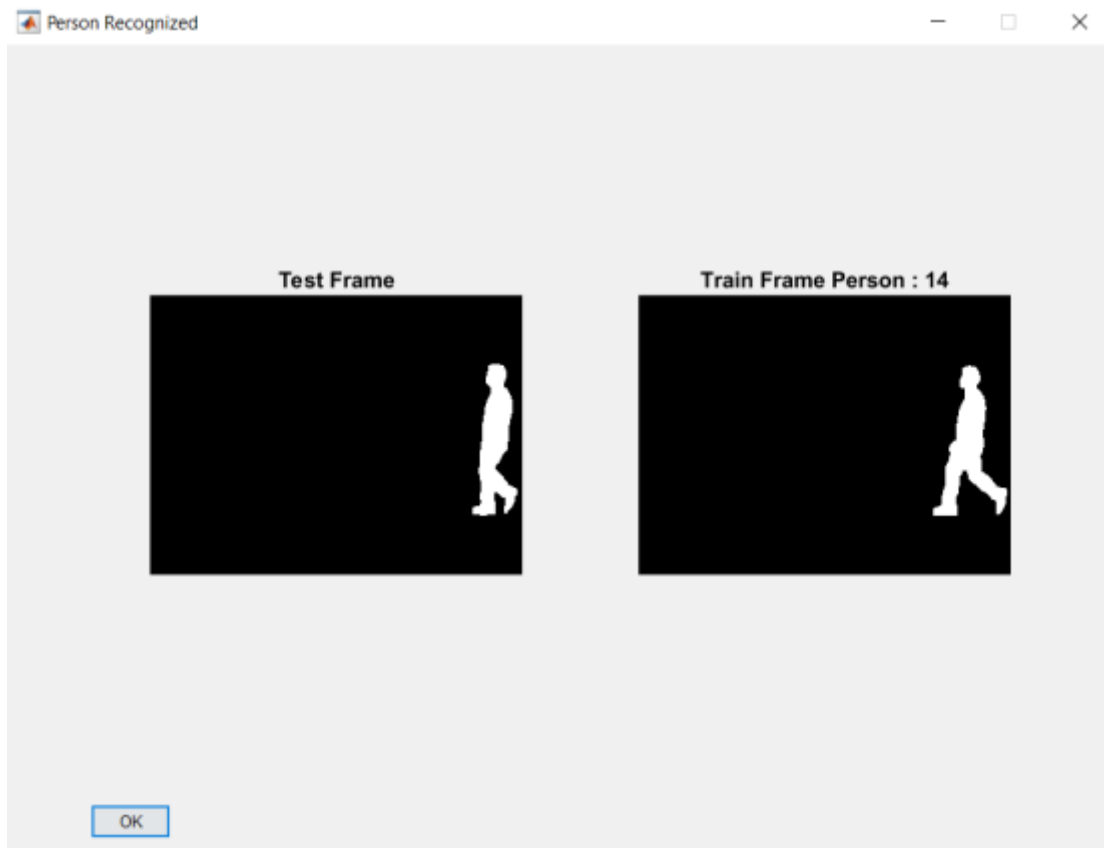


Εικόνα 4.2. 15 Operation Completed

Αν το άτομο έχει αναγνωριστεί, σαν τίτλος θα εμφανιστεί «Person Recognized». Αν όχι το μήνυμα θα είναι «Person Was Not Recognized , Please Try Again». Στη δεύτερη περίπτωση απλά ξαναεπιλέγουμε άλλο άτομο ή απλά διαφορετική γωνία.

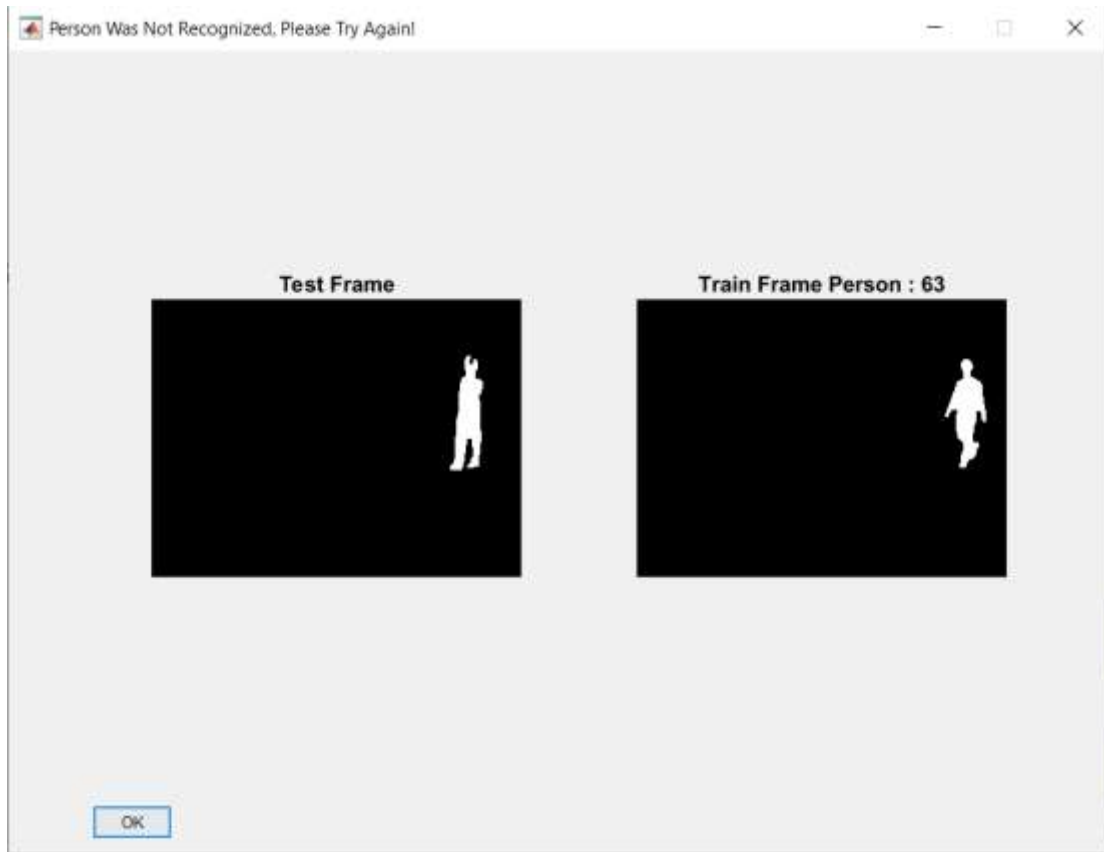
Σαν αποτέλεσμα εμφανίζονται δύο εικόνες. Η αριστερή εικόνα είναι το πρώτο καρέ από το άτομο το οποίο έχουμε επιλέξει να συγκρίνουμε (Test) και η δεξιά εικόνα είναι πρώτο καρέ του ατόμου το οποίο έχει αναγνωριστεί (Train).

Στην περίπτωση επιτυχούς αναγνώρισης βλέπουμε πως το Train άτομο είναι ίδιο με το Test, που είναι και το επιθυμητό.



Εικόνα 4.2. 16 Person Recognized

Στην περίπτωση, όμως, που το άτομο δεν έχει αναγνωριστεί με επιτυχία, στη δεξιά εικόνα δεν εμφανίζεται ο ίδιος αριθμός ατόμου. Εμφανίζεται ο αριθμός του ατόμου, που βρέθηκε ως πλησιέστερο στο Test αλλά δεν ταιριάζει με τον αριθμό του ατόμου που επιλέξαμε να συγκρίνουμε.



Εικόνα 4.2. 17 Person was not Recognized

Μπορούμε να δοκιμάσουμε όσες συγκρίσεις θέλουμε και τελειώνοντας κλείνουμε την εφαρμογή πατώντας το κουμπί «Exit».

## ΚΕΦΑΛΑΙΟ 5

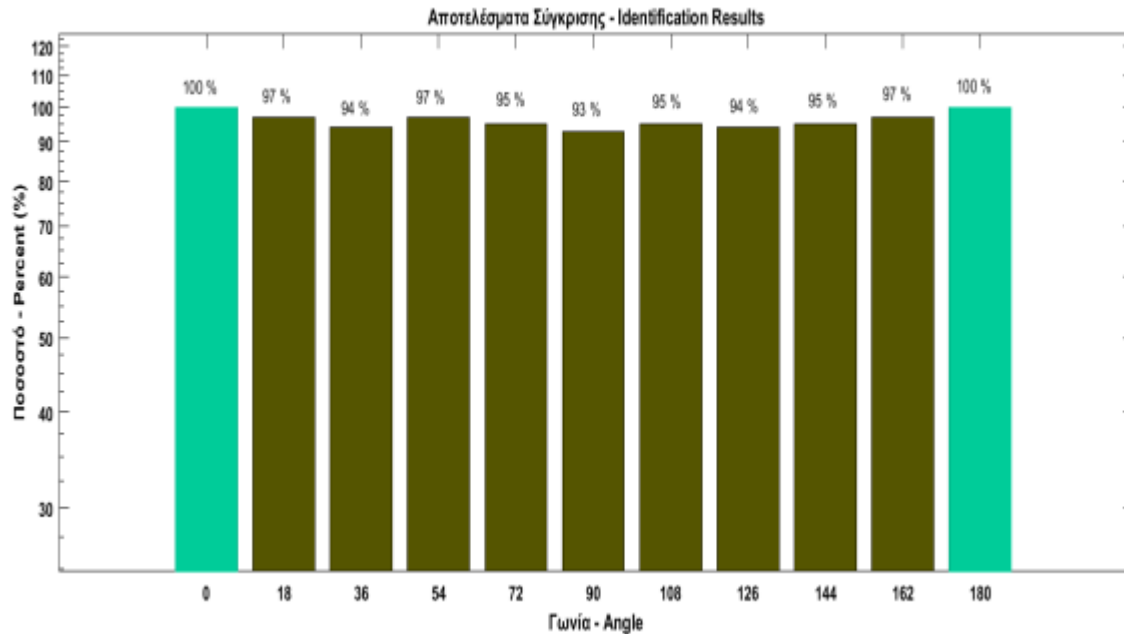
### 5.1 Συμπεράσματα - Αποτελέσματα

Τελικά Αποτελέσματα Επιτυχίας Αναγνώρισης Ανθρώπινου Βαδίσματος

<b>ΓΩΝΙΑ</b>	<b>ΠΟΣΟΣΤΟ (%)</b>
<b>0</b>	100
<b>18</b>	97
<b>36</b>	94
<b>54</b>	97
<b>72</b>	95
<b>90</b>	93
<b>108</b>	95
<b>126</b>	94
<b>144</b>	95
<b>162</b>	97
<b>180</b>	100

Πίνακας 5.1. 1 Πίνακας Αποτελεσμάτων

Συγκρίνοντας τα άτομα με την ευκλείδεια απόσταση βλέπουμε πως παίρνουμε αρκετά υψηλά αποτελέσματα. Παρατηρούμε πως στις γωνίες 0 και 180 τα άτομα έχουν αναγνωριστεί όλα με επιτυχία. Αυτό συμβαίνει γιατί σε αυτές τις γωνίες η στάση σώματος παραμένει σχεδόν ίδια και δεν αλλάζει.



Εικόνα 5.1 1 Bar Graph Results

Χρόνος Επεξεργασίας Δεδομένων

Χαρακτηριστικά Υπολογιστή: (RAM: 8 GB, CPU: AMD A12-9720p)

Έκδοση Matlab : R2016a

Για όλα τα άτομα:

AGDI : 15 λεπτά

Εξαγωγή Χαρακτηριστικών : 5 δευτερόλεπτα.

Ο χρόνος επεξεργασίας δεδομένων διαφέρει ανάλογα με τις επιδόσεις κάθε υπολογιστή, οπότε μπορούμε να έχουμε σαφώς καλύτερο χρόνο.

## 5.2 Μελλοντικές Έρευνες και Επεκτάσεις του Προγράμματος

Το πρόγραμμα που έχουμε υλοποιήσει μπορούμε να το επεκτείνουμε με ποικίλους τρόπους. Μπορούμε να εκτελέσουμε πειράματα με περισσότερα δεδομένα από τη βάση, ώστε να βγάλουμε πιο ασφαλή αποτελέσματα. Όπως αναφέρεται πιο πάνω, η βάση δεδομένων που χρησιμοποιούμε έχει 3 ειδών ακολουθίες. Εμείς χρησιμοποιούμε μόνο τη μία από τις τρεις, τη normal walking. Θα μπορούσαμε να δοκιμάσουμε με την ίδια διαδικασία να συγκρίνουμε τις σιλουέτες με παλτό (in a coat cl-01/02) και με τσάντα (with a bag bg-01/02).

Επίσης, αν χρησιμοποιούσαμε έγχρωμες εικόνες (χωρίς την απαλοιφή φόντου), θα ήταν ενδιαφέρον να συγκρίνουμε το ποσοστό αναγνώρισης σε σχέση με τις ασπρόμαυρες. Μια ενδιαφέρουσα οπτική, θα ήταν αν υπήρχαν διαφορετικές καιρικές συνθήκες ή μειωμένη ορατότητα στις φιγούρες.

Τέλος, για περαιτέρω ακρίβεια και αξιοπιστία θα απαιτούνταν η σύγκριση διαφορετικών γωνιών. Να μπορεί να αναγνωρίσει ένα άτομο όχι μόνο στην ίδια γωνία, που υλοποιούμε εμείς, αλλά και σε οποιαδήποτε άλλη γωνία.

Για να έχουμε ένα πιο ολοκληρωμένο και ακριβές πρόγραμμα αναγνώρισης ανθρώπινου βαδίσματος θα πρέπει να γίνουν αρκετά και διαφορετικά πειράματα.



## ΒΙΒΛΙΟΓΡΑΦΙΑ / ΑΝΑΦΟΡΕΣ

- [1] M. W. Whittle, “Clinical gait analysis: a review,” *Human Movement Science*, vol. 15, no. 3, pp. 369–387, 1996
- [2] S. Lobet, C. Detrembleur, F. Massaad, and C. Hermans, “Three-dimensional gait analysis can shed new light on walking in patients with haemophilia,” *The Scientific World Journal*, vol. 2013, Article ID 284358, 7 pages, 2013
- [3] S. V. Stevenage, M. S. Nixon, and K. Vince, “Visual analysis of gait as a cue to identity,” *Applied Cognitive Psychology*, vol. 13, no. 6, pp. 513–526, 1999
- [4] C. Ben Abdelkader, R. Cutler, and L. Davis, “Stride and cadence as a biometric in automatic person identification and verification,” in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, 2002
- [5] D. Cunado, M. S. Nixon, and J. N. Carter, “Automatic extraction and description of human gait models for recognition purposes,” *Computer Vision and Image Understanding*, vol. 90, no. 1, pp. 1–41, 2003
- [6] J. H. Yoo, D. Hwang, and M. S. Nixon, “Gender classification in human gait using support vector machine,” in *Advanced Concepts for Intelligent Vision Systems: 7th International Conference, ACIVS 2005, Antwerp, Belgium, September 20-23, 2005. Proceedings*, vol. 3708 of *Lecture Notes in Computer Science*, pp. 138–145, Springer, 2005
- [7] L. Lee and W. E. L. Grimson, “Gait analysis for recognition and classification,” in *Proceedings of 5th IEEE International Conference on Automatic Face Gesture Recognition*, pp. 155–162, 2002
- [8] C. Y. C. Yam, M. S. Nixon, and J. N. Carter, “Automated person recognition by walking and running via model-based approaches,” *Pattern Recognition*, vol. 37, no. 5, pp. 1057–1072, 2004
- [9] C. Yam, M. S. Nixon, and J. N. Carter, “Gait recognition by walking and running: a model-based approach,” in *Proceedings of the 5th Asian Conference on Computer Vision*, Melbourne, Australia, 2002

- [10] F. Tafazzoli and R. Safabakhsh, "Model-based human gait recognition using leg and arm movements," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1237–1246, 2010
- [11] M.-H. Cheng, M.-F. Ho, and C.-L. Huang, "Gait analysis for human identification through manifold learning and HMM," *Pattern Recognition*, vol. 41, no. 8, pp. 2541–2553, 2008
- [12] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, "Factorial HMM and parallel HMM for gait recognition," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 39, no. 1, pp. 114–123, 2009
- [13] A. Kale, A. Roy-Chowdhury, and R. Chellappa, "Gait-based human identification from a monocular video sequence," in *Handbook on Pattern Recognition and Computer Vision*, C. H. Cheng and P. S. P. Wang, Eds., World Scientific Publishing Company, 3rd edition, 2005
- [14] H. Murase and R. Sakai, "Moving object recognition in eigenspace representation: gait analysis and lip reading," *Pattern Recognition Letters*, vol. 17, no. 2, pp. 155–162, 1996
- [15] J. J. Little and J. E. Boyd, "Recognizing people by their gait: the shape of motion," *Videre: Journal of Computer Vision Research*, vol. 1, no. 2, 1998
- [16] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003
- [17] C. P. Lee, A. W. C. Tan, and S. C. Tan, "Gait recognition via optimally interpolated deformable contours," *Pattern Recognition Letters*, vol. 34, no. 6, pp. 663–669, 2013
- [18] H. Hu, "Enhanced gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1274–1286, 2013
- [19] S. Hong, H. Lee, and E. Kim, "Probabilistic gait modelling and recognition," *Computer Vision*, vol. 7, no. 1, pp. 56–70, 2013
- [20] C. Wang, J. Zhang, L. Wang, J. Pu, and X. Yuan, "Human identification using temporal information preserving gait template," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2164–2176, 2012

- [21] R. T. Collins, R. Gross, and S. Jianbo, "Silhouette-based human identification from body shape and gait," in Proceedings of 5th IEEE International Conference on Automatic Face Gesture Recognition, 2002
- [22] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanID gait challenge problem: data sets, performance, and analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 162–177, 2005
- [23] J. Chen, "Gait correlation analysis based human identification," The Scientific World Journal, vol. 2014, Article ID 168275, 8 pages, 2014
- [24] S. Yu, T. Tan, K. Huang, K. Jia, and X. Wu, "A study on gait-based gender classification," IEEE Transactions on Image Processing, vol. 18, no. 8, pp. 1905–1910, 2009
- [25] J. Han, B. Bhanu, "Individual Recognition using gait energy image", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, p. 316-322, Feb 2006
- [26] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, "Frame difference energy image for gait recognition with incomplete silhouettes," Pattern Recognition Letters, vol. 30, no. 11, pp. 977–984, 2009
- [27] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 131–137, 2004
- [28] "CASIA GAIT Database" 2009, <http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp>
- [29] "CASIA GAIT Database" 2009, [http://www.cbsr.ia.ac.cn/users/szheng/?page\\_id=71](http://www.cbsr.ia.ac.cn/users/szheng/?page_id=71)
- [30] Γεωργιάδης Κωνσταντίνος, "Ανάπτυξη Λογισμικού Ανίχνευσης Προσώπων σε Πολυμέσα", Κεφάλαιο 1, σελ. 15-16, Πτυχιακή Εργασία, ΑΤΕΙ Καβάλας, 2011

## ΠΑΡΑΡΤΗΜΑ Α΄

Πληροφορίες Ατόμων Αναγνώρισης

<b>Subject</b>	<b>ID</b>	<b>Gender</b>	<b>Height(cm)</b>
1	1	M	176
2	3	M	161
3	4	M	168
4	6	M	169
5	7	F	164
6	8	F	152
7	9	M	172
8	10	M	173
9	11	M	173
10	12	F	154
11	13	M	188
12	14	M	167
13	15	M	186
14	17	M	171
15	18	M	192
16	19	M	174
17	21	F	169
18	25	M	179
19	26	M	182
20	27	M	179
21	29	M	175
22	30	M	181
23	31	F	166
24	32	M	182
25	35	M	178
26	38	M	164
27	39	M	173
28	40	M	161
29	41	M	172
30	42	M	183
31	43	F	164
32	44	F	166
33	45	M	170
34	47	F	160
35	49	M	179
36	50	M	178

37	52	M	173
38	56	M	176
39	57	M	176
40	59	M	176
41	60	F	170
42	61	M	168
43	62	M	178
44	63	M	171
45	66	M	172
46	69	M	173
47	71	M	182
48	72	F	157
49	73	M	168
50	75	M	184
51	78	M	178
52	79	M	176
53	80	M	173
54	82	F	170
55	84	M	166
56	85	M	182
57	86	M	176
58	87	M	175
59	89	M	182
60	90	M	183
61	91	M	175
62	92	M	181
63	93	M	171
64	95	M	179
65	97	F	159
66	99	M	181
67	100	F	164
68	102	F	161
69	103	M	175
70	104	F	161
71	105	M	177
72	106	M	177
73	107	F	169
74	108	F	167
75	110	F	156
76	111	M	175
77	113	M	178
78	114	M	172

<b>79</b>	115	M	175
<b>80</b>	117	M	175
<b>81</b>	118	F	163
<b>82</b>	119	M	174
<b>83</b>	121	F	168
<b>84</b>	122	M	181
<b>85</b>	123	M	165
<b>86</b>	124	M	168

Πίνακας Α. 1 Πληροφορίες Ατόμων Αναγνώρισης

## ΠΑΡΑΡΤΗΜΑ Β΄

### Κώδικας Εφαρμογής

Ολοκληρωμένος κώδικας της εφαρμογής μας που υλοποιήσαμε στο περιβάλλον της Matlab.

```
function varargout = AGDI_GUI(varargin)
% AGDI_GUI MATLAB code for AGDI_GUI.fig
%   AGDI_GUI, by itself, creates a new AGDI_GUI or raises the existing
%   singleton*.
%
%   H = AGDI_GUI returns the handle to a new AGDI_GUI or the handle to
%   the existing singleton*.
%
%   AGDI_GUI('CALLBACK', hObject,eventData,handles,...) calls the local
%   function named CALLBACK in AGDI_GUI.M with the given input arguments.
%
%   AGDI_GUI('Property','Value',...) creates a new AGDI_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before AGDI_GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to AGDI_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help AGDI_GUI

% Last Modified by GUIDE v2.5 18-Oct-2020 00:57:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @AGDI_GUI_OpeningFcn, ...
                  'gui_OutputFcn', @AGDI_GUI_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before AGDI_GUI is made visible.
function AGDI_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to AGDI_GUI (see VARARGIN)

% Choose default command line output for AGDI_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes AGDI_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = AGDI_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in AGDI.
function AGDI_Callback(hObject, eventdata, handles)
% hObject    handle to AGDI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% Part 1 - The construction of AGDI

agdi = zeros(86, 11, 160, 90);

```



```

folder = uigetdir();
persons = dir(folder);
h = waitbar(0,'Please wait, calculations in progress...');
for iPerson = 3:length(persons)
    angles = dir(fullfile(folder, persons(iPerson).name, 'nm-01'));
    for iAngle = 3:length(angles)
        filename = fullfile(folder, persons(iPerson).name, 'nm-01', angles(iAngle).name);

        % store AGDI for each person and angle
        agdi(iPerson-2, iAngle-2, :, :) = reshape(AGDI(filename), 1, 1, 160, 90);
    end
    waitbar(0.33,h, 'Please wait, calculations in progress...');
    pause(0.001)

end
waitbar(0.5,h, 'Please wait, calculations in progress...');
pause(0.001)

save('agdi.mat', 'agdi');
setappdata(handles.AGDI, 'folder', folder);
setappdata(handles.AGDI, 'persons', persons);
setappdata(handles.AGDI, 'angles', angles);
setappdata(handles.AGDI, 'agdi', agdi);
waitbar(1,h,'Finishing');
pause(0.001)
close(h);
set(handles.AGDI,'string','AGDI','BackgroundColor','green','enable','off');

% --- Executes on button press in Feature_Extraction.
function Feature_Extraction_Callback(hObject, eventdata, handles)
% hObject    handle to Feature_Extraction (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% Part 2 - Feature Extraction
h=waitbar(0, 'Please wait, calculations in progress...');
pause(0.001)

agdi = getappdata(handles.AGDI, 'agdi');
angles = getappdata(handles.AGDI, 'angles');
persons = getappdata(handles.AGDI, 'persons');

m_agdi = reshape(mean(agdi, 1), 11, 160, 90); % mean AGDI for each angle

```

```

Ct = cell(11, 1);
for iAngle = 1:11
    for iPerson = 1:86
        % diff = G - E(G) from eq(7)
        diff = reshape(agdi(iPerson, iAngle, :, :), 160, 90) - reshape(m_agdi(iAngle, :, :), 160, 90);
        if (all(size(Ct{iAngle}) == [0 0]))
            Ct{iAngle} = diff * diff;
        else
            Ct{iAngle} = Ct{iAngle} + diff * diff;
        end
    end
end
waitbar(0.33,h, 'Please wait, calculations in progress...');
pause(0.001)

end
waitbar(.5,h,'Please wait, calculations in progress...');
pause(0.001)

k = 20;
features = cell(86, 11);
for iAngle = 1:11
    for iPerson = 1:86
        % store features for each person and each angle
        features{iPerson, iAngle} = feature_extraction(reshape(agdi(iPerson, iAngle, :, :), [160
90]), Ct{iAngle}, k);
    end
    waitbar(.75,h, 'Please wait, calculations in progress...');
    pause(0.001)
end

setappdata(handles.Feature_Extraction, 'Ct', Ct);
setappdata(handles.Feature_Extraction, 'k', k);
setappdata(handles.Feature_Extraction, 'features', features);
%--

pe = [1:length(persons)-2]';
set(handles.choose_person, 'String', num2str(pe));

an=[000:018:180]';
set(handles.choose_angle, 'String', an);
waitbar(1,h,'Finishing');
pause(0.001)
close(h);

```

```
set(handles.Feature_Extraction,'string','Feature  
Extraction','BackgroundColor','green','enable','off');
```

```
% --- Executes on selection change in choose_person.
```

```
function choose_person_Callback(hObject, eventdata, handles)
```

```
% hObject handle to choose_person (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns choose_person contents as cell array
```

```
% contents{get(hObject,'Value')} returns selected item from choose_person
```

```
% --- Executes during object creation, after setting all properties.
```

```
function choose_person_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to choose_person (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on selection change in choose_angle.
```

```
function choose_angle_Callback(hObject, eventdata, handles)
```

```
% hObject handle to choose_angle (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns choose_angle contents as cell array
```

```
% contents{get(hObject,'Value')} returns selected item from choose_angle
```

```
% --- Executes during object creation, after setting all properties.
```

```
function choose_angle_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to choose_angle (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in identify.
function identify_Callback(hObject, eventdata, handles)
% hObject    handle to identify (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
folder = getappdata(handles.AGDI, 'folder');
Ct = getappdata(handles.Feature_Extraction, 'Ct');
k = getappdata(handles.Feature_Extraction, 'k');
features = getappdata(handles.Feature_Extraction, 'features');

```

```

P = get(handles.choose_person,'value')+2;
A = get(handles.choose_angle,'value')+2;

```

```

persons = dir(folder);
for iPerson = P
    angles = dir(fullfile(folder, persons(iPerson).name, 'nm-02'));
    for iAngle = A
        % Extract features for each person and each angle from test set
        filename = fullfile(folder, persons(iPerson).name, 'nm-02', angles(iAngle).name);
        agdi = AGDI(filename);
        Y = feature_extraction(agdi, Ct{iAngle-2}, k);

        % Get the distances from the features of all person on this angle
        % from train set
        d = 100 * ones(86, 1);
        for jPerson = 1:86
            d(jPerson) = dist(Y, features{jPerson, iAngle-2});
        end
    end
end

```

```

% Match is the person with the closer distance
[~, match] = min(d);

```

```

filename2 = fullfile(folder, persons(match+2).name, 'nm-01', angles(iAngle).name);
bn = dir(filename2);
bn2 = fullfile(filename2, bn(3).name);

```

```

D2=imread(bn2);

an = dir(filename);
an2 = fullfile(filename, an(3).name);
D=imread(an2);

% If the match person is the original person -> success
if (match == iPerson-2)
    h=msgbox('Operation Completed','Person Recognized');
    set(h, 'position', [400 100 550 400]); %makes box bigger
    subplot(1,2,1);imshow(D);
    title ('Test Frame')
    subplot(1,2,2);imshow(D2);
    title (['Train Frame' ' Person : ' num2str(match)])

else
    h=msgbox('Operation Completed','Person Was Not Recognized, Please Try Again!');
    set(h, 'position', [400 100 550 400]); %makes box bigger
    subplot(1,2,1);imshow(D);
    title ('Test Frame')
    subplot(1,2,2);imshow(D2);
    title (['Train Frame' ' Person : ' num2str(match)] )

end
end
end

% --- Executes on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% hObject    handle to exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close

% --- Executes on button press in view_silhouettes.
function view_silhouettes_Callback(hObject, eventdata, handles)
% hObject    handle to view_silhouettes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```
% handles structure with handles and user data (see GUIDATA)
```

## VIEW\_IMAGES

```
function varargout = VIEW_IMAGES(varargin)
% VIEW_IMAGES MATLAB code for VIEW_IMAGES.fig
% VIEW_IMAGES, by itself, creates a new VIEW_IMAGES or raises the existing
% singleton*.
%
% H = VIEW_IMAGES returns the handle to a new VIEW_IMAGES or the handle to
% the existing singleton*.
%
% VIEW_IMAGES('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in VIEW_IMAGES.M with the given input arguments.
%
% VIEW_IMAGES('Property','Value',...) creates a new VIEW_IMAGES or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before VIEW_IMAGES_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to VIEW_IMAGES_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help VIEW_IMAGES

% Last Modified by GUIDE v2.5 04-Jul-2021 20:01:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @VIEW_IMAGES_OpeningFcn, ...
    'gui_OutputFcn', @VIEW_IMAGES_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before VIEW_IMAGES is made visible.
function VIEW_IMAGES_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to VIEW_IMAGES (see VARARGIN)

% Choose default command line output for VIEW_IMAGES
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes VIEW_IMAGES wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = VIEW_IMAGES_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in open_file.
function open_file_Callback(hObject, eventdata, handles)
% hObject    handle to open_file (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
folder = uigetdir();
persons = dir(folder);

```

```

setappdata(handles.open_file, 'folder', folder);
setappdata(handles.open_file, 'persons', persons);

% --- Executes on selection change in train_test.
function train_test_Callback(hObject, eventdata, handles)
% hObject    handle to train_test (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns train_test contents as cell array
%         contents{get(hObject,'Value')} returns selected item from train_test
persons = getappdata(handles.open_file, 'persons');

str = get(handles.train_test,'string');
val_train_test = get(handles.train_test,'value');

te = [1:(length(persons)-2)];
set(handles.person, 'String', num2str(te));

v=[000:018:180]';
set(handles.angle, 'String', v);

setappdata(handles.train_test, 'val_train_test', val_train_test);

% --- Executes during object creation, after setting all properties.
function train_test_CreateFcn(hObject, eventdata, handles)
% hObject    handle to train_test (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in person.
function person_Callback(hObject, eventdata, handles)
% hObject    handle to person (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```



```

% Hints: contents = cellstr(get(hObject,'String')) returns person contents as cell array
%     contents{get(hObject,'Value')} returns selected item from person
val_person = get(handles.person,'value');
setappdata(handles.person, 'val_person', val_person);

% --- Executes during object creation, after setting all properties.
function person_CreateFcn(hObject, eventdata, handles)
% hObject    handle to person (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in angle.
function angle_Callback(hObject, eventdata, handles)
% hObject    handle to angle (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns angle contents as cell array
%     contents{get(hObject,'Value')} returns selected item from angle
str_angle = get(handles.angle,'string');
val_angle = get(handles.angle,'value');
setappdata(handles.angle, 'val_angle', val_angle);
setappdata(handles.angle, 'str_angle', str_angle);

% --- Executes during object creation, after setting all properties.
function angle_CreateFcn(hObject, eventdata, handles)
% hObject    handle to angle (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in view_person.
function view_person_Callback(hObject, eventdata, handles)
% hObject   handle to view_person (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
folder = getappdata(handles.open_file, 'folder');
persons = getappdata(handles.open_file, 'persons');
val_person = getappdata(handles.person, 'val_person');
val_train_test = getappdata(handles.train_test, 'val_train_test');
val_angle = getappdata(handles.angle, 'val_angle');

switch val_train_test
    case 1
        view_f1=fullfile(folder, persons(val_person+2).name, 'nm-01');
    case 2
        view_f1=fullfile(folder, persons(val_person+2).name, 'nm-02');
end
view_f2=dir(view_f1);
view_f3=fullfile(view_f1, view_f2(val_angle+2).name);
view_f4=dir(view_f3);
fr=3;
D=imread(fullfile(view_f3,view_f4(fr).name));
imshow(D);

set(handles.Frame_number,'String',num2str(fr-2));
set(handles.Length_Frame,'String',num2str(length(view_f4)-2));

setappdata(handles.view_person, 'fr', fr);
setappdata(handles.view_person, 'view_f3', view_f3);
setappdata(handles.view_person, 'view_f4', view_f4);

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over angle.
function angle_ButtonDownFcn(hObject, eventdata, handles)
% hObject   handle to angle (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% --- Executes on button press in previous.
function previous_Callback(hObject, eventdata, handles)

```

```

% hObject  handle to previous (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
view_f3=getappdata(handles.view_person, 'view_f3');
view_f4=getappdata(handles.view_person, 'view_f4');
fr=getappdata(handles.view_person, 'fr');
fr=fr-1;
set(handles.Frame_number,'String',num2str(fr-2));
if fr>3
D=imread(fullfile(view_f3,view_f4(fr).name));
imshow(D);
setappdata(handles.view_person, 'fr', fr);
end
% --- Executes on button press in next.
function next_Callback(hObject, eventdata, handles)
% hObject  handle to next (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
view_f3=getappdata(handles.view_person, 'view_f3');
view_f4=getappdata(handles.view_person, 'view_f4');
fr=getappdata(handles.view_person, 'fr');

if fr<(length(view_f4))
    fr=fr+1;
D=imread(fullfile(view_f3,view_f4(fr).name));
imshow(D);
set(handles.Frame_number,'String',num2str(fr-2));
setappdata(handles.view_person, 'fr', fr);
end

% --- Executes on button press in close_viewer.
function close_viewer_Callback(hObject, eventdata, handles)
% hObject  handle to close_viewer (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
close VIEW_IMAGES

% --- Executes during object creation, after setting all properties.
function Frame_number_CreateFcn(hObject, eventdata, handles)
% hObject  handle to Frame_number (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

```

```
% --- Executes during object creation, after setting all properties.  
function Length_Frame_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to Length_Frame (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```

## ΠΑΡΑΡΤΗΜΑ Γ΄

### Κώδικας Προγράμματος

Ολοκληρωμένος κώδικας του πρόγρραμματος για τα τελικά αποτελέσματα.

#### %% Part 1 - The construction of AGDI

```
agdi = zeros(86, 11, 160, 90);

folder = uigetdir();
persons = dir(folder);
for iPerson = 3:length(persons)
    angles = dir(fullfile(folder, persons(iPerson).name, 'nm-01'));
    for iAngle = 3:length(angles)
        filename = fullfile(folder, persons(iPerson).name, 'nm-01', angles(iAngle).name);

        % store AGDI for each person and angle
        agdi(iPerson-2, iAngle-2, :, :) = reshape(AGDI(filename), 1, 1, 160, 90);
    end
end
```

```
save('agdi.mat', 'agdi');
% clear filename iAngle iPerson
```

#### %% Part 2 - Feature Extraction

```
m_agdi = reshape(mean(agdi, 1), 11, 160, 90); % mean AGDI for each angle
Ct = cell(11, 1);
for iAngle = 1:11
    for iPerson = 1:86
        % diff = G - E(G) from eq(7)
        diff = reshape(agdi(iPerson, iAngle, :, :), 160, 90) - reshape(m_agdi(iAngle, :, :), 160, 90);
        if (all(size(Ct{iAngle}) == [0 0]))
            Ct{iAngle} = diff' * diff;
        else
            Ct{iAngle} = Ct{iAngle} + diff' * diff;
        end
    end
end
end
```

```

k = 20;
features = cell(86, 11);
for iAngle = 1:11
    for iPerson = 1:86
        % store features for each person and each angle
        features{iPerson, iAngle} = feature_extraction(reshape(agdi(iPerson, iAngle, :, :), [160
90]), Ct{iAngle}, k);
    end
end

% clear iPerson iAngle diff

%% Part 3 - Identification
score = zeros(11, 1);

persons = dir(folder);
for iPerson = 3:length(persons)
    angles = dir(fullfile(folder, persons(iPerson).name, 'nm-02'));
    for iAngle = 3:length(angles)
        % Extract features for each person and each angle from test set
        filename = fullfile(folder, persons(iPerson).name, 'nm-02', angles(iAngle).name);
        agdi = AGDI(filename);
        Y = feature_extraction(agdi, Ct{iAngle-2}, k);

        % Get the distances from the features of all person on this angle
        % from train set
        d = 100 * ones(86, 1);
        for jPerson = 1:86
            d(jPerson) = dist(Y, features{jPerson, iAngle-2});
        end

        % Match is the person with the closer distance
        [~, match] = min(d);

        % If the match person is the original person -> success (+1)
        if (match == iPerson-2)
            score(iAngle-2) = score(iAngle-2) + 1;
        end
    end
end

angle = [0 18 36 54 72 90 108 126 144 162 180]';
score = score / 86 * 100;

```

```
table(angle, score)
```

```
% clear persons angles iPerson iAngle filename folder jPerson
```

```
function adgi = AGDI(dirname)
```

```
% AGDI - Returns the Average Gait Differential Image of the images
```

```
% contained in the folder dirname
```

```
filenames = dir(dirname);
```

```
frames = cell(length(filenames)-2, 1);
```

```
centroids = cell(length(filenames)-2, 1);
```

```
% Traverse frames of person's gait
```

```
for i = 3:length(filenames)
```

```
    % Load BW image
```

```
    img = logical(imread(fullfile(dirname, filenames(i).name)));
```

```
    % Crop black background
```

```
    bb = regionprops(img, 'BoundingBox');
```

```
    img = imcrop(img, bb(end).BoundingBox);
```

```
    % Resize
```

```
    img = imresize(img, [160, 90], 'bicubic');
```

```
    % Centroid
```

```
    centroid = regionprops(img, 'Centroid');
```

```
    frames{i-2} = img;
```

```
    centroids{i-2} = centroid.Centroid;
```

```
end
```

```
dgi = cell(length(filenames)-3, 1);
```

```
adgi = zeros(160, 90);
```

```
for i = 1:length(filenames)-3
```

```
    dgi{i} = DGI(frames{i}, frames{i+1}, centroids{i}, centroids{i+1}); % compute DGI
```

```
    dgi{i} = imresize(dgi{i}, [160 90], 'bicubic'); % resize to 160x90
```

```
    adgi = adgi + dgi{i};
```

```
end
```

```
adgi = adgi / (length(filenamees)-3);
```

```
end
```

```
function difference = DGI(img1, img2, centroid1, centroid2)
```

```
% DGI - Finds the Differential Gait Image
```

```
assert(all(size(img1) == size(img2)));
```

```
% How much each image should be padded to perform alignment in the centroid
```

```
pad = double(int8(abs(centroid1 - centroid2)));
```

```
% Pad images so the centroids get aligned
```

```
if centroid1(1) < centroid2(1)
```

```
    img1 = padarray(img1, [pad(1) 0], 'post');
```

```
    img2 = padarray(img2, [pad(1) 0], 'pre');
```

```
else
```

```
    img1 = padarray(img1, [pad(1) 0], 'pre');
```

```
    img2 = padarray(img2, [pad(1) 0], 'post');
```

```
end
```

```
if centroid1(2) < centroid2(2)
```

```
    img1 = padarray(img1, [0 pad(2)], 'post');
```

```
    img2 = padarray(img2, [0 pad(2)], 'pre');
```

```
else
```

```
    img1 = padarray(img1, [0 pad(2)], 'pre');
```

```
    img2 = padarray(img2, [0 pad(2)], 'post');
```

```
end
```

```
% Differential Gait Image
```

```
difference = abs(img1 - img2);
```

```
end
```

```
function Y = feature_extraction(adgi, Ct, d)
```

```
% FEATURE_EXTRACTION extracts d features from the ADGI given the image
```

```
% covariance matrix Ct
```

```
% Extract d largest eigenvectors
```



```

[W, ~] = eig(Ct);
W = W(:, end-d+1:end);

Y = adgi * W; % Yk = GWk for k = 1, 2, ..., d (in matrix form)

end

```

```

function d = dist(Y1, Y2)
% Computes the distance of 2 features matrices

d = 0;
for i = 1:size(Y1, 2)
    d = d + norm(Y1(:, i) - Y2(:, i));
end

end

```

```

function agdi = reconstruction(Y, Ct)
% RECONSTRUCTION - Reconstructs the ADGI image given its features Y, and
% the image covariance matrix Ct

% Find feature dimensionality
d = size(Y, 2);

% Extract d largest eigenvectors
[W, ~] = eig(Ct);
W = W(:, end-d+1:end);

% agdi = sum(Yk*Wk') as in equation (17)
agdi = zeros(160, 90);
for i = 1:d
    agdi = agdi + Y(:, i) * W(:, i)';
end

end

```