ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εξαγωγή χαρακτηριστικών καρέ από ακολουθίες βίντεο με παίκτες Super Mario

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΔΗΜΗΤΡΙΟΥ ΣΩΤΗΡΟΠΟΥΛΟΥ

Επιβλέπων: Καθ. Στέφανος Κόλλιας

Αθήνα, Ιούνιος 2016

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εξαγωγή χαρακτηριστικών καρέ από ακολουθίες βίντεο με παίκτες Super Mario

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΔΗΜΗΤΡΙΟΥ ΣΩΤΗΡΟΠΟΥΛΟΥ

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2016

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....	.....	.....
Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης	Δρ.Κωνσταντίνος Καρπούζης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Ερευνητής Ε.Μ.Π., ΙΕΕΕ

Δημήτριος Σωτηρόπουλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Στέφανο Κόλλια που είχε την επίβλεψη της συγκεκριμένης διπλωματικής εργασίας και τον δρα. Κωνσταντίνο Καρπούζη για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον και σύγχρονο αντικείμενο καθώς και για την αμέριστη βοήθειά του καθ'όλη τη διάρκεια της παρούσας μελέτης.

Περίληψη

Η εξαγωγή χαρακτηριστικών καρέ από μια ακολουθία βίντεο είναι ένας σημαντικός ερευνητικός και αναπτυξιακός τομέας, αφού μας δίνει τη δυνατότητα να παράγουμε γρήγορα και αυτόματα μια σύνοψη ενός βίντεο με λίγες εικόνες ή να το χαρακτηρίσουμε με βάση λέξεις-κλειδιά που μας ενδιαφέρουν. Στις περισσότερες περιπτώσεις, αυτό γίνεται χρησιμοποιώντας χαμηλού επιπέδου χαρακτηριστικά, όπως το χρώμα ή η κίνηση και αναζητώντας καρέ με την ελάχιστη συσχέτιση με τα υπόλοιπα (ώστε να είναι, ουσιαστικά, διαφορετικά από τα υπόλοιπα στην ακολουθία).

Στη συγκεκριμένη διπλωματική εργασία, επεξεργαζόμαστε βίντεο από ανθρώπους που παίζουν Super Mario, σε συνδυασμό με τα δεδομένα από το ίδιο το παιχνίδι, ώστε να εντοπίσουμε στιγμιότυπα από κάθε βίντεο, τα οποία να δίνουν ένδειξη του αν ο παίκτης είναι ικανοποιημένος ή όχι από τη συγκεκριμένη πίστα, αν αυτή του προκαλεί εκνευρισμό ή βαρεμάρα και αν το επίπεδο πρόκλησης είναι αρκετό, ώστε να μπορέσουμε να παράγουμε με δυναμικό/συναρτησιακό τρόπο πίστες που να έχουν αρκετές πιθανότητες να ταιριάζουν με τις ικανότητες και τη διάθεση του συγκεκριμένου παίκτη. Παράγοντας τα χαρακτηριστικά καρέ, χρησιμοποιούμε στρατηγικές crowdsourcing για να πάρουμε χαρακτηρισμούς για το τι δείχνει καθένα από αυτά από ανώνυμους χρήστες και χρησιμοποιούμε αυτούς του χαρακτηρισμούς για να προβλέψουμε την εμπειρία παίκτη σε κάθε περίπτωση.

Λέξεις Κλειδιά: <<επεξεργασία βίντεο, χαρακτηριστικά καρέ, βιντεοπαιχνίδια, μοντελοποίηση εμπειρίας παίκτη, εξατομίκευση περιεχομένου, οπτικές αντιδράσεις>>

Abstract

Extracting feature frames from a video sequence is a very important sector in research and development, since it makes it possible for us to quickly and automatically produce a video synopsis consisting of a few pictures or characterize it based on index terms that are of interest to us. On most occasions, this is achieved by using low level features, like color or movement and searching for frames of minimum correlation to the rest (so that they essentially are different to the rest of the sequence).

In this thesis, we are processing videos from people that are playing Super Mario, combined with data from the game itself, in order to find instances in each video that indicate whether the player is satisfied with that particular level or not, if the level causes frustration or boredom and if the challenge level is sufficient, so that we can produce levels that have a good chance to fit the mood and skills of each player, in a dynamic/functional way. After extracting the characteristic frames, we are using crowdsourcing techniques to obtain characterizations on what each one of them (from anonymous users) shows and we use these characterizations to predict player experience in each case.

Index Terms: <<Video processing, characteristic frames, videogames, player experience modeling, content personalization, visual cues.**>>**

Πίνακας Περιεχομένων

1. Εισαγωγή

Η βιομηχανία των βιντεοπαιχνιδιών βρίσκεται σε άνθιση εδώ και τουλάχιστον τρεις δεκαετίες, με έσοδα που πολλές φορές ξεπερνούν τις βιομηχανίες των ταινιών και της μουσικής. Χάρη στην υψηλή δημοτικότητα αλλά και τις τεράστιες υπολογιστικές ανάγκες τα βιντεοπαιχνίδια πάντα παρουσιάζουν τεχνολογίες αιχμής και πρωτοποριακές μεθόδους στον τομέα της αλληλεπίδρασης ανθρώπου-υπολογιστή. Η σημερινή τεχνολογία έχει φθάσει σε ένα επίπεδο τέτοιο ώστε νέα addons να μπορούν να εμπλουτίσουν το gameplay αλλοιώνοντας και κατευθύνοντας το περιεχόμενο και την εξέλιξη του παιχνιδιού ακολουθώντας στρατηγικές εξαρτώμενες από την επιρροή του παιχνιδιού στον παίκτη. Για το σκοπό αυτό, η χρήση παραμέτρων σχετικών με το γενικότερο πλαίσιο και τη συμπεριφορά για την εξαγωγή πληροφοριών σχετικών με την τρέχουσα κατάσταση του παίκτη είναι πρωταρχικής σημασίας για την κατασκευή μοντέλων προσωπικής συμπεριφοράς και σχετικών με την αλληλεπίδραση και την καθοδήγηση προσαρμογής του παιχνιδιού ώστε να επιτευχθεί η μέγιστη δυνατή ευχαρίστηση/προσήλωση.

Υπάρχουν πολλές μελέτες σχετικές με την εκτίμηση της κατάστασης του χρήση κατά την αλληλεπίδραση ανθρώπου-υπολογιστή. Πρόφατη πρόοδος σε τεχνικές οράσεως υπολογιστών υπό μη ελεγχόμενες συνθήκες έχουν επιτρέψει την πρόταση τεχνικών που εμπεριέχουν έννοιες όπως κινήσεις των ματιών, του κεφαλιού και του σώματος και εκφράσεις του προσώπου. Ειδικότερα στο πεδίο των βιντεοπαιχνιδιών, η αυξημένη διαφοροποίηση δημογραφικά και όσον αφορά στρατηγικές, ανάγκες, ικανότητες και προτιμήσεις των παικτών έχει αυξήσει τη σημασία της εξατομίκευσης της εμπειρίας. Μελέτες της μοντελοποίησης της εμπειρίας παικτών που βασίζονται σε λεπτομέρειες ενεργειών του χρήστη, παρέχουν κάποια αρχικά σημεία αναφοράς για την επίτευξη ενός τέτοιου στόχου.

Η μέτρηση φυσιολογικών σημάτων είναι ένα δημοφιλές εργαλείο σε αυτό το πλαίσιο, για να επιτευχθεί όμως απαιτεί τη χρήση εξειδικευμένου εξοπλισμού, ο οποίος πολύ συχνά είναι ακριβός και δύσκολος να βαθμονομηθεί. Σαν αποτέλεσμα, οι τεχνικές αυτές ίσως είναι αποτελεσματικές όσον αφορά την αναγνώριση της επίδρασης στον παίκτη αλλά πολύ προβληματικές για ανάπτυξη σε μεγάλη κλίμακα ή για εμπορικούς σκοπούς. Αφ' ετέρου, προσεγγίσεις βασισμένες στην επεξεργασία ακολουθιών βίντεο από χαμηλών προδιαγραφών κάμερες (όπως οι κάμερες του Kinect για το Xbox της Microsoft ή μια απλή κάμερα για τον υπολογιστή) χρησιμοποιούν υλικό το οποίο βρίσκεται ήδη στην κατοχή των περισσοτέρων παικτών και δεν επιβάλλουν πρόσθετες απαιτήσεις όπως η κίνηση σε περιορισμένο χώρο.

Μια άλλη κατεύθυνση που λαμβάνει αυξανόμενη προσοχή είναι η δυναμική παραγωγή περιεχομένου και η δημιουργία εξατομικευμένου υλικού για τον παίκτη ή το σχεδιαστή ως αποτέλεσμα. Το πρώτο βήμα για την παραγωγή εξατομικευμένου περιεχομένου είναι η αποτελεσματική μοντελοποίηση της σχέσης μεταξύ του περιεχομένου και της εμπειρίας του παίκτη. Αυτό μπορεί να επιτευχθεί κατασκευάζοντας μοντέλα πάνω σε δεδομένα των οποίων η συλλογή γίνεται κατά τη διάρκεια της αλληλεπίδρασης του χρήστη με το ψηφιακό υλικό.

Στην εργασία [1] που αποτέλεσε βάση για τη διπλωματική εργασία, χρησιμοποιήθηκε ένα σύστημα σύντηξης παραμέτρων του παιχνιδιού, δεικτών απόδοσης και ενός συνόλου οπτικών χαρακτηριστικών του κεφαλιού του παίκτη με σκοπό την πρόβλεψη των προτιμήσεων του παίκτη μεταξύ διαφόρων μεταβλητών του παιχνιδιού. Συλλέχθηκε ένα μεγάλο σώμα δεδομένων συμπεριφοράς και οπτικών ερεθισμάτων, καθώς και σχόλια για την προσωπική εμπειρία του καθενός, από 58 χρήστες ενώ έπαιζαν το δημοφιλές platform παιχνίδι “Super Mario Bros”. Οι υποκειμενικές αναφορές των παικτών αναγνωρίσθηκαν μέσω συγκριτικών ερωτηματολογίων και οι διάφορες

μεταβλητές του παιχνιδιού κατατάσσονται σε σχέση με το βαθμό πρόκλησης, απορρόφησης και εκνευρισμού. Τα αποτελέσματα έδειξαν ότι μπορούν να κατασκευαστούν μοντέλα υψηλής ακριβείας καθώς έφθασαν σε 91%, 92% και 88% ακρίβεια για απορρόφηση/προσήλωση, εκνευρισμό και πρόκληση αντίστοιχα. Τα μοντέλα αυτά χρησιμοποιούνται για την παραγωγή εξατομικευμένων επιπέδων για τους παίκτες.

1.1 Αντικείμενο Προηγούμενης Συγγενικής Μελέτης

1.1.1 Παιχνίδι-Πλατφόρμα δοκιμών

Εικ 1: Στιγμιότυπο από το παιχνίδι "Infinite Mario Bros"

Το παιχνίδι platform που χρησιμοποιήθηκε για τις δοκιμές της μελέτης είναι μια τροποποιημένη εκδοχή του “Infinite Mario Bros” του Markus Persson, το οποίο είναι ένας κοινού κτήματος κλώνος του κλασικού παιχνιδιού platform της Nintendo, “Super Mario Bros”. Το αυθεντικό “Infinite Mario Bros” και ο πηγαίος κώδικάς του είναι διαθέσιμα στο διαδίκτυο*.

Το gameplay του Super Mario Bros αποτελείται από την κίνηση του ελεγχομένου από τον παίκτη χαρακτήρα, Μάριο, σε επίπεδα δύο διαστάσεων (2D). Ο Μάριο μπορεί να περπατά, να τρέχει, να σκύβει, να πηδάει και να εκτοξεύει πύρινες σφαίρες. Ο βασικός στόχος του κάθε επιπέδου είναι να φτάσεις στο τέλος του. Δευτερεύοντες στόχοι περιλαμβάνουν τη συλλογή όσο το δυνατόν περισσοτέρων νομισμάτων και την ολοκλήρωση του επιπέδου όσο το δυνατόν γρηγορότερα.

Ενώ το Infinite Mario Bros παρέχει τα περισσότερα από τα χαρακτηριστικά του Super Mario Bros, αυτό που το κάνει να ξεχωρίζει είναι η αυτόματη παραγωγή επιπέδων. Κάθε φορά που ξεκινά καινούριο παιχνίδι, τα επίπεδα δημιουργούνται τυχαία. Η τροποποιημένη εκδοχή επικεντρώθηκε σε κάποιες επιλεγμένες παραμέτρους που επηρεάζουν την εμπειρία του gameplay.

1.1.2 Σχεδιασμός Συνόλου Δεδομένων

Για την εκτίμηση της συναισθηματικής κατάστασης των παικτών κατά τη διάρκεια του παιχνιδιού σχεδιάστηκε το εξής πρωτόκολλο πειράματος. Πενήντα οκτώ εθελοντές (ηλικιών 22-48 ετών, 28 άντρες, 30 γυναίκες) κάθησαν μπροστά σε μια οθόνη υπολογιστή για εγγραφή βίντεο. Τα πειράματα έγιναν στην Ελλάδα και τη Δανία. Οι συνθήκες φωτισμού ήταν οι τυπικές περιβάλλοντος γραφείου και για τη σύλληψη της οπτικής συμπεριφοράς των παικτών χρησιμοποιήθηκε κάμερα υψηλής ευκρίνειας (Canon Legria S11). Για τη συλλογή αναφορών σχετικών με την προσωπική, υποκειμενική εμπειρία των παικτών χρησιμοποιήθηκε η εξής διαδικασία:

1) Μια εισαγωγή παρουσιάζει το παιχνίδι στον παίκτη και περιέχει πληροφορίες για τη διαδικασία που θα ακουλουθήσει. Ο παίκτης ενημερώνεται ότι κατά τη διάρκεια της συνεδρίας θα παίξει δύο σύντομα παιχνίδια και θα κληθεί να απαντήσει ερωτήσεις για την εμπειρία του.
2) Κατόπιν παρουσιάζεται ένα ερωτηματολόγιο που συλλέγει τις εξής πληροφορίες: ηλικία, πόσο συχνά παίζει ο χρήστης βιντεοπαιχνίδια, πόσο χρόνο την εβδομάδα αφιερώνει σε αυτά (0, 1, 2 ή περισσότερο από 3 ώρες την εβδομάδα) και το αν έχει ξαναπαίξει Super Mario Bros.
3) Ο παίκτης ενημερώνεται για το πώς θα χειρίζεται το Μάριο.
4) Ο παίκτης πληροφορείται ότι οι συνεδρίες καταγράφονται σε βίντεο και θα αναλυθούν.
5) Μετά από αυτά τα εισαγωγικά βήματα ο παίκτης είναι έτοιμος να ξεκινήσει το πρώτο παιχνίδι (παιχνίδι Α). Του δίνονται τρεις ευκαρίες να ολοκληρώσει ένα σύντομο επίπεδο του Super Mario Bros. Αν η πρώτη προσπάθεια αποτύχει το παιχνίδι επανεκκινείται στο αρχικό σημείο και ο παίκτης είναι έτοιμος να ξαναδοκιμάσει. Το παιχνίδι τελειώνει είτε ολοκληρώνοντας μια από τις τρεις δοκιμές είτε αποτυγχάνοντας στην τρίτη.
6) Μετά το τέλος του παιχνιδιού Α παρουσιάζεται ένα ερωτηματολόγιο στον παίκτη (LIKERT*). Ο παίκτης καλείται να εκφράσει τις συναισθηματικές του προτιμήσεις για το παιχνίδι σε κάθε μια από τις συναισθηματικές καταστάσεις (απορρόφηση, εκνευρισμό και πρόκληση). Το ερωτηματολόγιο είναι εμπνευσμένο από το ερωτηματολόγιο εμπειρίας παιχνιδιού (Game Experience Questionnaire, GEQ), σύμφωνα με το οποίο μια κλίμακα likert από το 0 μέχρι το 4 αντιπροσωπεύει τη δύναμη του συναισθήματος (4 σημαίνει επακρώς, 0 σημαίνει καθόλου).
7) Ένα δεύτερο σύντομο παιχνίδι (παιχνίδι Β) παρουσιάζεται στον παίκτη και είναι έτοιμος να παίξει. Ο παίκτης έχει τρεις ευκαιρίες (δηλαδή ζωές του Μάριο) να ολοκληρώσει το επίπεδο και εφαρμόζονται οι ίδιοι κανόνες με το παιχνίδι Α.
8) Μετά το τέλος του παιχνιδιού Β παρουσιάζεται το ερωτηματολόγιο GEQ στον παίκτη (όπως στο παιχνίδι Α).
9) Μετά την ολοκλήρωση των παιχνιδιών Α και Β ο παίκτης καλείται να αναφέρει το παιχνίδι που προτιμά για τις τρεις συναισθηματικές διαστάσεις μέσω ενός πρωτοκόλλου ερωτηματολογίων αναγκαστικής τετραπλής εναλλακτικής επιλογής (four-alternative forced choice, 4-AFC), οι επιλογές του οποίου είναι: προτιμάται το Α από το Β, προτιμάται το Β από το Α, και τα 2 προτιμούνται το ίδιο, δεν προτιμάται κανένα. Το ερωτηματολόγιο είναι της ακόλουθης μορφής: Ποιο παιχνίδι ήταν περισσότερο x, όπου x μια από τις τρεις συναισθηματικές καταστάσεις που ερευνούνται.
10) Ο παίκτης έχει την επιλογή να λήξει τη συνεδρία ή να συνεχίσει. Στη δεύτερη περίπτωση, παρουσιάζεται ένα νέο ζεύγος παιχνιδιών και η διαδικασία επαναλαμβάνεται.

Κάθε συμμετέχων έπαιξε από δύο έως πέντε ζευγάρια παιχνιδιών κατά μέσο όρο, με αποτέλεσμα ένα σύνολο από 380 παιχνίδια (περισσότερο από έξι ώρες καταγεγραμμένων βίντεο). Στις περισσότερες περιπτώσεις, ο παίκτης ήταν μόνος του στο δωμάτιο όσο έπαιζε και όταν αυτό δεν ήταν δυνατόν, ζητήθηκε να μην τον αποσπά κανένας. Οι συνεδρίες του παιχνιδιού που παρουσιάστηκαν στους παίκτες κατασκευάστηκαν χρησιμοποιώντας πλάτος επιπέδου 100 μονάδων Super Mario Bros (κουτάκια). Η επιλογή του μεγέθους αυτού έγινε λόγω συμβιβασμού μεταξύ ενός μεγέθους παραθύρου αρκετά μεγάλου για να επιτρέψει επαρκή αλληλεπίδραση μεταξύ του παίκτη και του παιχνιδιού ώστε να ενεργοποιηθούν οι υπό εξέταση συναισθηματικές καταστάσεις και ενός παραθύρου αρκετά μικρού για να θέσει μια αποδεκτή συχνότητα ενός μηχανισμού προσαρμογής εφηρμοσμένου σε πραγματικό χρόνο με σκοπό το κλείσιμο του συναισθηματικού βρόχου του παιχνιδιού.

Μετά την αφαίρεση των στιγμών των συνεδριών για τις οποίες τα οπτικά δεδομένα ήταν κατεστραμμένα, το πλήρες σύνολο δεδομένων αποτελείται από 167 ζεύγη παιχνιδιών. Επιπροσθέτως, εφαρμόστηκε ένα βήμα προεπεξεργασίας για να αφαιρεθούν τα ζεύγη παιχνιδιών για τα οποία οι παίκτες ανέφεραν ασαφείς προτιμήσεις (τα παιχνίδια που προτιμούσαν ή δεν προτιμούσαν ισότιμα).

1.1.3 Εξαγωγή Χαρακτηριστικών

Οι υποπαράγραφοι που ακολουθούν περιγράφουν τα χαρακτηριστικά που είχαν εξαχθεί και χρησιμοποιηθεί στη συγγενική εργασία [1] ως προγνωστικοί παράγοντες της αναφερομένης εμπειρίας. Περιλαμβάνονται χαρακτηριστικά των επιπέδων του παιχνιδιού, χαρακτηριστικά συμπεριφοράς gameplay, και χαρακτηριστικά κινήσεως κεφαλιού. Το τμήμα κλείνει με την περιγραφή των σχολιασμών της εμπειρίας του παίκτη.

Α. Χαρακτηριστικά Περιεχομένου

Η γεννήτρια επιπέδων του παιχνιδιού είχε τροποποιηθεί για να δημιουργεί σύμφωνα με τα επόμενα έξι ελεγχόμενα (περιεχόμενο παιχνιδιού) χαρακτηριστικά:

1) τον αριθμό των κενών στο επίπεδο, G,

2) το μέσο πλάτος των κενών, G_w,

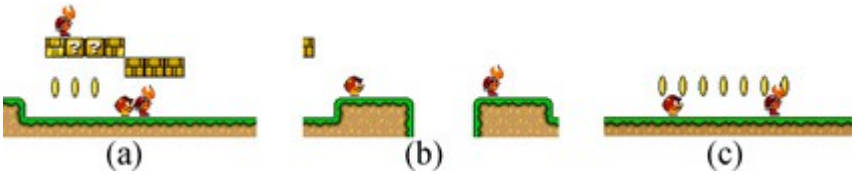
3) τον αριθμό των εχθρών E. Αυτή η παράμετρος ελέγχει τον αριθμό των χελωνών και goompas που είναι σκορπισμένοι σε όλο το επίπεδο και αλλάζουν το βαθμό δυσκολίας,

4) την τοποθέτηση των εχθρών E_P . Ο τρόπος που τοποθετούνται οι εχθροί στο επίπεδο αποφασίζεται από τρεις πιθανότητες οι οποίες προστίθενται σε μία:

a) Γύρω από οριζόντια κουτιά, P_b: οι εχθροί τοποθετούνται πάνω σε ή κάτω από ένα σύνολο οριζοντίων κουτιών (μια ακολουθία κουτιών τοποθετημένων οριζόντια χωρίς να εφάπτονται στο έδαφος.

b) Γύρω από κενά, P_g: οι εχθροί τοποθετούνται κοντά στην άκρη ενός κενού.

c) Τυχαία τοποθέτηση P_r: οι εχθροί τοποθετούνται σε ένα επίπεδο σημείο στο έδαφος.



Εικ 2: Τοποθέτηση εχθρών χρησιμοποιώντας διαφορετικές πιθανότητες. Δίνεται υψηλή πιθανότητα στην τοποθέτηση γύρω από οριζόντια κουτιά. (a) Pb. (b) Γύρω από κενά, Pg. (c) Τυχαία, Pr

Η Εικ 2 απεικονίζει τοποθετημένους εχθρούς δίνοντας διαφορετικές τιμές στις P_b, P_g και P_r. Η εικόνα 2(a) δείχνει εχθρούς τοποθετημένους σύμφωνα με την ανάθεση της τιμής 80% στην P_b. Η 2(b) απεικονίζει το αποτέλεσμα την ανάθεσης της τιμής 80% στην P_g. Η 2(c) έιναι αποτέλεσμα της P_r = 80%.

5) τον αριθμό των powerups, N_w. Ο Μάριο μπορεί να συλλέξει στοιχεία powerups κρυμμένα σε κουτιά για να αναβαθμιστεί από μικρό σε μεγάλο ή από μεγάλο σε φωτιά.

6) τον αριθμό των κουτιών, B. Ορίζουμε μια μεταβλητή για να προσδιορίσουμε τον αριθμό των δύο διαφορετικών τύπων κουτιών που υπάρχουν στο Super Mario. Καλούμε αυτά τα κουτιά μπλοκ και βράχους. Τα μπλοκς περιέχουν κρυφά στοιχεία όπως νομίσματα και powerups. Οι βράχοι μπορεί να κρύβουν ένα νόμισμα, ένα powerup, ή να είναι άδειοι. Ο Μάριο μπορεί να σπάσει τους βράχους μόνο όταν είναι μεγάλος.

Σύμφωνα με τη μεθοδολογία του [5], δύο καταστάσεις (υψηλή και χαμηλή) ορίζονται για κάθε μία από τις παραπάνω ελεγχόμενες παραμέτρους εκτός από την τοποθέτηση των εχθρών, στην οποία έχουν ανατεθεί τρεις διαφορετικές καταστάσεις, επιτρέποντας με αυτόν τον τρόπο περισσότερο έλεγχο πάνω στη δυσκολία και την ποικιλία των δημιουργημένων επιπέδων. Η επιλογή των συγκεκριμένων ελεγχομένων χαρακτηριστικών έγινε λαμβάνοντας υπόψιν τις συμβουλές ειδικών σε σχεδιασμό βιντεοπαιχνιδιών, και με την πρόθεση να καλυφθούν τα χαρακτηριστικά που έχουν τη μεγαλύτερη επίδραση στις υπό έρευνα συναισθηματικές καταστάσεις. [4], [5]

Β. Χαρακτηριστικά Gameplay

Όσο παιζόταν το παιχνίδι, καταγράφηκαν διάφορες δράσεις των παικτών και αλληλεπιδράσεις με αντικείμενα του παιχνιδιού και τις αντίστοιχες χρονικές στιγμές.Αυτά τα συμβάντα κατηγοριοποιήθηκαν σε διάφορες ομάδες ανάλογα με τον τύπο του συμβάντος και τον τύπο της αλληλεπίδρασης με τα αντικείμενα του παιχνιδιού. Τα συμβάντα που καταγράφηκαν είναι τα ακόλουθα: συμβάν ολοκλήρωσης επιπέδου, συμβάν θανάτου Μάριο και αίτιο θανάτου, συμβάντα αλληλεπίδρασης με αντικείμενα του παιχνιδιού όπως νομίσματα, άδειοι βράχοι, βράχοι/μπλοκς νομισμάτων και βράχοι/ μπλοκς powerups, συμβάν όπου ο Μάριο σκοτώνει έναν εχθρό μαζί με τον τύπο του εχθρού και τις πράξεις με τις οποίες το σκότωσε, συμβάν αλλαγής μορφής του Μάριο (μικρός, μεγάλος ή φωτιά), συμβάν αλλαγής κατάστασης του Μάριο (μετακίνηση δεξιά, αριστερά, άλμα, τρέξιμο, σκύψιμο) και η πλήρης τροχιά του Μάριο ως συνδυασμός συμβάντων.

Αρκετά χαρακτηριστικά εξήχθησαν άμεσα από τα δεδομένα που καταγράφηκαν. Τα περισσότερα από αυτά τα χαρακτηριστικά εμφανίζονται σε προηγούμενες μελέτες [4]-[6] και η επιλογή τους έγινε έτσι ώστε να μπορούν να αντιπροσωπεύσουν τις διαφορές ανάμεσα σε μια μεγάλη ποικιλία από τρόπους παιξίματος του Super Mario Bros.

Γ. Χαρακτηριστικά Κίνησης Κεφαλιού

Στα πειράματα της [1], όσο οι παίκτες κάθονταν μπορστά από μια οθόνη υπολογιστή, το πάνω μέρος του σώματός τους παρακολουθείτο από μια κάμερα, ενώ η κίνηση του κεφαλιού ήταν ιδιαίτερης σημασίας για τη δημιουργία συσχετίσεων συμπεριφοράς με συμβάντα του παιχνιδιού και επίπεδα δυσκολίας και διαφορετικά πρότυπα κινήσεων συσχετίζονταν με την κατάσταση και τις προτιμήσεις διαφορετικών παικτών. Παραδείγματος χάριν, παρατηρήθηκε ότι ο εκνευρισμός συνδέθηκε με απότομες και πολύ γρήγορες κινήσεις του κεφαλιού, ενώ χαμηλά επίπεδα πρόκλησης συνήθως θα σχετίζονταν με πιο απαλές κινήσεις, πιθανότατα εξαιτίας έλλειψης υψηλού ενδιαφέροντος. Για τους παραπάνω λόγους, εξετάστηκε η σχέση μιας σειράς από χαρακτηριστικά κινήσεων του κεφαλιού με μοντέλα εμπειρίας, καθώς και με χαρακτηριστικά gameplay και περιεχομένου. Πιο συγκεκριμένα, η κίνηση του κεφαλιού του παίκτη ανιχνεύθηκε μέσω οριζοντίων και καθέτων (στροφή/yaw και κλίση/pitch) περιστροφικών κινήσεων. Αυτές εξήχθησαν με τρόπους που ευνόησαν την υπολογιστική πολυπλοκότητα, ακρίβεια και ευρωστία σε διάφορες συνθήκες φωτισμού και αυθόρμητες κινήσεις. Οι τιμές των εξηχθέντων χαρακτηριστικών λαμβάνονται υπόψιν τόσο κατά την διάρκεια των συνεδριών του παιχνιδιού (χαρακτηριστικά μέσης κίνησης κεφαλιού) όσο και σε μικρές περιόδους κρίσιμων συμβάντων (χαρακτηριστικά οπτικής αντίδρασης).

1) *Χαρακτηριστικά Μέσης Κίνησης Κεφαλιού:* Εδώ θεωρήθηκε ως κίνηση κεφαλιού η πρώτη παράγωγος του τύπου του διανύσματος στάσης κεφαλιού και χρησιμοποιήθηκε ο μέσος όρος (Avg) των απολύτων τιμών της για τη διάρκεια των συνεδριών του παιχνιδιού. Μια ακολουθία από περεταίρω χαρακτηριστικά κινήσεων του κεφαλιού λήθφηκαν υπόψιν με σκοπό την απόσπαση συναισθηματικών πληροφοριών από τον παίκτη κατά τη διάρκεια κάθε συνεδρίας (Χαρακτηριστικά Μέσης Κίνησης Κεφαλιού). Πιο συγκεκριμένα λήφθηκαν υπόψιν:

a) η Συνολική Ενεργοποίηση (OA), η οποία προκύπτει ως το άθροιμα των ποσοτήτων της κίνησης για κάθε περιστροφική κίνηση ξεχωριστά. Με άλλα λόγια, η OA αντιπροσωπεύει την ποσότητα κίνησης για συγκεκριμένες χρονικές περιόδους. Έστω H μια ακολουθία από νύξεις στάσης κεφαλιού για την αντίστοιχη συνεδρία, αποτελούμενη από T καρέ, όπως στην 1

$$H = [(y_1^H, p_1^H), (y_2^H, p_2^H), \ldots, (y_T^H, p_T^H)] \quad (1)$$

όπου y_i^H, p_i^H είναι οι απόλυτες γωνίες στροφής και κλίσης (yaw και pitch) αντίστοιχα. Η Ολική Ενεργοποίηση της Στάσης Κεφαλιού για την ακολουθία H είναι

$$OA = \sum_{i=1}^{T} (dYaw + dPitch) \quad (2)$$

όπου

$$dYaw = dy/dt \quad (3)$$

και

$$dPitch = dp/dt \quad (4)$$

b) η παράμετρος της Χρονικής Εκφραστικότητας (TE), η οποία δηλώνει την ταχύτητα της κίνησης και διαχωρίζει γρήγορες και αργές κινήσεις του κεφαλιού, είναι η μέση τιμή της OA για χρονική περίοδο T.

c) η παράμετρος της Χωρικής Έκτασης (SE) είναι η μέγιστη τιμή της στιγμιαίας παρέκλισσης από μια μετωπική θέση (y=p=0).

Category	Feature	Description
		Content (Level) Features
Content (Level) Features	G	Number of gaps
	$\bar{G}_w$	Average width of gaps
	E	Number of enemies
	E_p	Placement of enemies
	N_w	Number of powerups
	B	Number of boxes
		GamePlay Features
Time	t_{comp}	Completion time
	$t_{lastLift}$	Playing duration of last life over total time spent on the level
	t_{duck}	Time spent ducking (%)
	t_{jump}	Time spent jumping (%)
	t_{left}	Time spent moving left (%)
	t_{right}	Time spent moving right (%)
	t_{run}	Time spent running (%)
	t_{small}	Time spent in Small Mario mode (%)
	t_{big}	Time spent in Big Mario mode (%)
Interaction with items	n_{coins}	Free coins collected (%)
	$n_{coinBlocks}$	Coin blocks pressed or coin rocks destroyed (%)
	$n_{powerups}$	Powerups pressed (%)
	n_{boxes}	Sum of all blocks and rocks pressed or destroyed (%)
Interaction with enemies	$k_{cannonFlower}$	Times the player kills a cannonball or a flower (%)
	$k_{goombaKoopa}$	Times the player kills a goomba or a koopa (%)
	k_{stomp}	Opponents died from stomping (%)
	$k_{unleash}$	Opponents died from unleashing a turtle shell (%)
Death	d_{total}	Total number of deaths
	d_{cause}	Cause of the last death
Miscellaneous	n_{mode}	Number of times the player shifted the mode between: Small, Big, and Fire
	n_{jump}	Number of times the jump button was pressed
	n_{gJump}	Difference between the # of gaps and the # of jumps
	n_{duck}	Number of times the duck button was pressed
	n_{state}	Number of times the player changed the state between: standing still, run, jump, moving left, and moving right

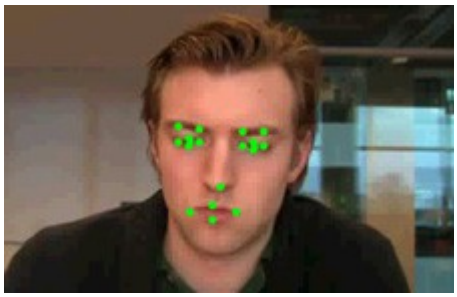
d) η παράμετρος της Εκφραστικότητας Ενέργειας (ή Δύναμη) της κίνησης του κεφαλιού (PO) κατά τη διάρκεια της φάσης του "χτυπήματος". Οι κινήσεις του κεφαλιού (όπως και οι χειρονομίες) θεωρείται ότι αποτελούνται από τρεις φάσεις: προετοιμασία, "χτύπημα" και ανάκληση. Το μήνυμα μεταφέρεται κυρίως κατά τη φάση του "χτυπήματος", ενώ οι φάσεις της προετοιμασίας και της ανάκλησης συμβαίνουν ενώ το κεφάλι κινείται από και προς τη φυσική του θέση, αντίστοιχα. Η τυποποίηση αυτής της παραμέτρου σύμφωνα με αυτόν τον ορισμό βέβαια, δεν είναι καθόλου υποτυπώδης, εφόσον ο αυτόματος εντοπισμός αυτών των σταδίων είναι μια αρκετά δύσκολη πρόκληση. Εναλλακτικά, η παράμετρος αυτή συσχετίστηκε ποιοτικά με την πρώτη παράγωγο της ταχύτητας (επιτάχυνση) σε συγκεκριμένες χρονικές περιόδους (5)

$$PO = [\Sigma_{i=1}^{T}(d^2y_i/dt^2 + d^2p_i/dt^2)]/T \quad (5)$$

e) η Ρευστότητα της κίνησης κεφαλιού (FL) κάνει το διαχωρισμό ανάμεσα σε ομαλές και απότομες κινήσεις. Υπό αυτό το πρίσμα, η μεταβολή της ταχύτητας εξετάστηκε για τα 2 μέρη της στάσης κεφαλιού που χρησιμοποιήθηκαν. Αυτή η έννοια επιχειρεί να δηλώσει τη συνέχεια των κινήσεων, ανεξάρτητα από το μέγεθος της ταχύτητας. Η σχέση (6) δείχνει τον υπολογισμό της παραμέτρου της ρευστότητας

$$FL = [var(dYaw) + var(dPitch)]/2 \quad (6)$$

Η παραπάνω ποσότητα έχει υψηλές τιμές για χρονικές περιόδους που περιέχουν απότομες/ αυθόρμητες/ απρόβλεπτες κινήσεις και χαμηλές τιμές για κινήσεις μεγαλύτερης συνέχειας.



Εικ. 3: Εντοπισμός χαρακτηριστικών του προσώπου για εξαγωγή χαρακτηριστικών κίνησης κεφαλιού.

Στην συγγενική εργασία [1] εκτός από τα παραπάνω χαρακτηριστικά, εξετάστηκαν και οι μέσες τιμές των οριζοντίων, $M_{horizontal}$, και καθέτων, $M_{vertical}$ περιστροφών, καθώς και οι μέσοι όροι των τύπων περιστροφής κεφαλιού M.

2) Χαρακτηριστικά Οπτικής Αντίδρασης: καθώς η εκφραστικότητα των παικτών φαίνεται να αυξάνεται κατά τη διάρκεια συγκεκριμένων γεγονότων, τα παραπάνω χαρακτηριστικά εξετάστηκαν σε σχέση με κάποια σημεία του gameplay, ως εξής:

a) όταν ο παίκτης χάσει μια ζωή,

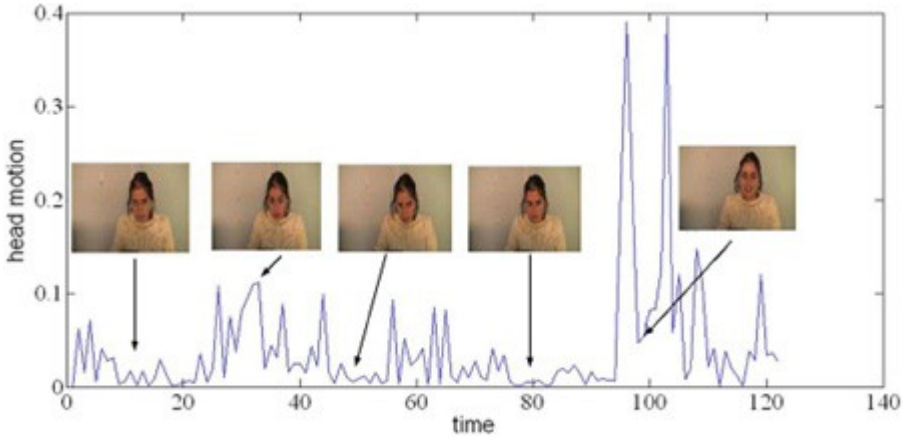
b) όταν ο παίκτης σκοτώσει έναν εχθρό πηδώντας πάνω του,

c) όταν ο παίκτης ξεκινά ή ολοκληρώνει μια κρίσιμη κίνηση: κάνει άλμα, σκύβει, τρέχει και κινείται δεξιά ή αριστερά,

d) όταν ο παίκτης αλληλεπιδρά με ένα αντικείμενο.

Αυτά τα χαρακτηριστικά υπολογίζονται για περιόδους δεκα καρέ πριν και μετά από τα αντίστοιχα συμβάντα. Στη συνέχεια συγκρίθηκαν οι μέσες τιμές του με τις αντίστοιχες μέσες τιμές κατά τη διάρκεια κανονικού gameplay, για κάθε συνεδρία ξεχωριστά.

Category	Feature	Description
		Head Movement Features throughout whole sessions
Mean Head Movement	Avg	Absolute first order derivative of Head Pose Vector
	OA	Overall Activation
	SE	Spatial Extent
	TE	Temporal Expressivity parameter
	PO	Energy Expressivity parameter
	FL	Fluidity
	$M_{horizontal}$	Median value for horizontal head rotation
	$M_{vertical}$	Median value for vertical head rotation
		Head Movement Features during gameplay events
Visual Reaction Features	Avg_a	Absolute first order derivative of Head Pose Vector when the gameplay event, a occur
	OA_a	Overall Activation when the gameplay event, a occur
	SE_a	Spatial Extent when the gameplay event, a occur
	TE_a	Temporal Expressivity parameter when the gameplay event, a occur
	PO_a	Energy Expressivity parameter when the gameplay event, a occur
	FL_a	Fluidity when the gameplay event, a occur
	M_a	Median value for head rotation norm when the gameplay event, a occur



Εικ. 4: Τυπική εκφραστικότητα κεφαλιού παίκτη που αντιδρά σε συγκεκριμένα συμβάντα στο παιχνίδι.

Δ. Εμπειρία του Παίκτη

Όπως αναφέρθηκε νωρίτερα, η εμπειρία του παίκτη μετράται μέσω ερωτηματολογίου με τέσσερις αναγκαστικές εναλλακτικές επιλογές, το οποίο παρουσιάζεται στον παίκτη αφού παίξει ένα ζεύγος παιχνιδιών δημιουργημένων από διαφορετικό σύνολο ελεγχομένων τιμών χαρακτηριστικών. Το ερωτηματολόγιο ζητά από τον παίκτη να αναφέρει το παιχνίδι που προτίμησε σε τρεις καταστάσεις χρήστη: προσήλωση, πρόκληση και εκνευρισμό. Η επιλογή των καταστάσεων αυτών βασίστηκε σε προηγούμενες μελέτες και έρευνες σχετικές με βιντεοπαιχνίδια [4], με σκοπό τη σύλληψη τόσο συναισθηματικών όσο και συμπεριφορικών στοιχείων της εμπειρίας gameplay.

1.1.4 Εκμάθηση Προτιμήσεων για την Μοντελοποίηση της Εμπειρίας Παικτών

Η νευρο-εξελισσόμενη εκμάθηση προτιμήσεων [7], [3] έχει χρησιμοποιηθεί για την κατασκευή μοντέλων που προσεγγίζουν τη λειτουργία μεταξύ gameplay, χαρακτηριστικών κίνησης κεφαλιού, χαρακτηριστικών περιεχομένου και αναφερόμενες συναισθηματικές προτιμήσεις. Στη νευρο-

εξελισσόμενη εκμάθηση προτιμήσεων ένας γενετικός αλγόριθμος (genetic algorithm, GA) εξελίσσει ένα τεχνητό νευρωνικό δίκτυο (artificial neural network, ANN) έτσι ώστε η έξοδός του να ταιρίαζει με τις προτιμήσεις στο σύνολο δεδομένων. Η είσοδος του ANN είναι ένα σύνολο χαρακτηριστικών που έχουν εξαχθεί από το σύνολο δεδομένων. Ο GA που εφαρμόζεται χρησιμοποιεί μια συνάρτηση καταλληλότητας, η οποία μετράει τη διαφορά ανάμεσα στις καταγεγραμμένες προτιμησεις και το σχετικό μέγεθος της εξόδου του μοντέλου.

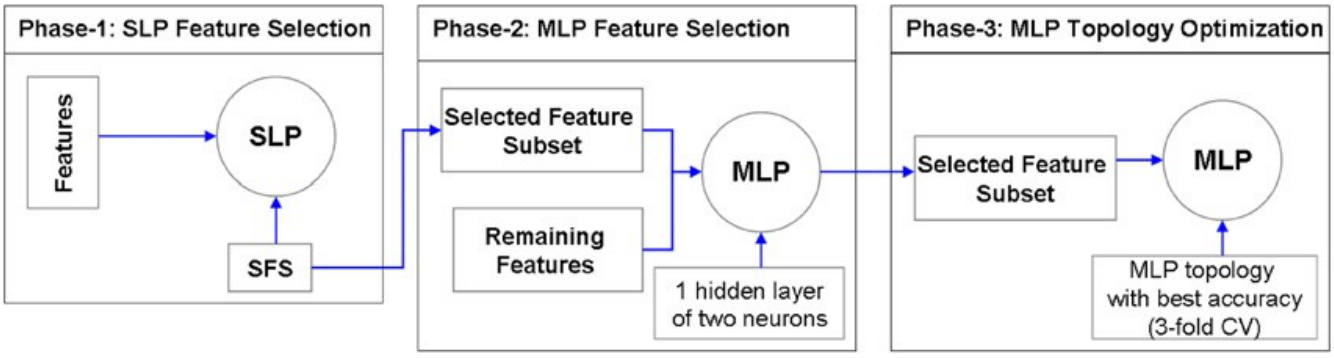
Όλα τα χαρακτηριστικά που εξάγονται κανονικοποιούνται ομοιόμορφα σε [0, 1] χρησιμοποιώντας το πρότυπο κανονικοποίησης μεγίστου-ελαχίστου. Μετά την κανονικοποίηση, οι τιμές αυτές χρησιμοποιήθηκαν σαν είσοδοι για επιλογή χαρακτηριστικών και βελτιστοποίηση του μοντέλου ANN. Η μοντελοποίηση περιλαμβάνει τα ακόλουθα τρία βήματα:

1) Επιλογή χαρακτηριστικών: χρησιμοποιήθηκε sequential forward selection (SFS) για την επιλογή του σχετικού υποσυνόλου χαρακτηριστικών για την πρόβλεψη κάθε συναισθηματικής κατάστασης. Αυτό επιτεύχθηκε εκπαιδεύοντας single-layer perceptrons (SLPs) σαν χαρτογράφηση μεταξύ επιλεγμένων χαρακτηριστικών και καταγεγραμμένων προτιμήσεων. Η SFS είναι μια bottom-up προσέγγιση, όπου ένα χαρακτηριστικό επιλέγεται για να προστεθεί στο τρέχον σύνολο επιλεγμένων χαρακτηριστικών, καθώς το νέο υποσύνολο χαρακτηριστικών πετυχαίνει τη μέγιστη δυνατή επίδοση. Η ποιότητα ενός υποσυνόλου χαρακτηριστικών καθορίζεται από τριπλά διασταυρωμένη επικύρωση σε αφανή δεδομένα.

2) Επέκταση χώρου χαρακτηριστικών: το υποσύνολο των χαρακτηριστικών που παράγεται από την πρώτη φάση χρησιμοποιήθηκε σαν είσοδος στα μοντέλα small multilayer perceptron (MLP) ενός κρυμμένου στρώματος δύο νευρώνων και η SFS επιλέγει επιπρόσθετα χαρακτηριστικά απο το εναπομείναν σύνολο χαρακτηριστικών κατά τη διάρκεια της εκπαίδευσης αυτών των μικρών MLPs.

3) Βελτιστοποίηση τοπολογίας: στην τελική φάση της διαδικασίας μοντελοποίησης, η τοπολογία των μοντέλων MLP βελτιστοποιείται χρησιμοποιώντας νευρο-εξελισσόμενη εκμάθηση προτιμήσεων. Η διαδικασία βελτιστοποίησης της τοπολογίας δικτύου ξεκινά με ένα μικρο MLP δύο κρυμμένων νευρώνων και η τοπολογία δικτύου σταδιακά αυξάνεται σε 2 κρυμμένα στρώματα αποτελούμενα από δέκα νευρώνες το καθένα.

Η ποιότητα ενός υποσυνόλου χαρακτηριστικών και η απόδοση κάθε MLP εξασφαλίζεται μέσω της μέσης ακριβείας ταξινόμησης σε τρία ανεξάρτητα τρεξίματα χρησιμοποιώντας τριπλά διασταυρωμένη επικύρωση σε δέκα εξελικτικές δοκιμές. Διηξήχθησαν δοκιμές συντονισμού παραμέτρων για προετοιμασία των τιμών τους για νευρο-εξελισσόμενη εκμάθηση προτιμήσεων που αποφέρουν την ύψιστη ακρίβεια και ελαχιστοποιούν την υπολογιστική προσπάθεια. Σαν αποτέλεσμα αυτής της διαδικασίας συντονισμού παραμέτρων, χρησιμοποιήθηκε ένα πλήθος 100 ατόμων και εφαρμόσθηκε η εξέλιξη για 20 γενιές. Χρησιμοποιήθηκε ένα πιθανοτικό σύστημα επιλογής με βάση την κατάταξη, με τα άτομα που βρίσκονται υψηλότερα στην κατάταξη να έχουν μεγαλύτερη πιθανότητα να επιλεγούν ως “γονείς”. Τέλος, η αναπαραγωγή γίνεται μέσω ομοιόμορφων συνδυασμών, και ακολουθείται από Γκαουσσιανή μετάλλαξη πιθανότητας 1%.



1.1.5 Πειράματα και Ανάλυση συγγενικής εργασίας

Στις ακόλουθες παραγράφους περιγράφονται τα πειράματα που διηξήχθησαν στο πλαίσιο της εργασίας [1] με σκοπό την κατασκευή και σύγκριση διαφορετικών μοντέλων εμπειρίας παίκτη που προέρχονται από τα χαρακτηριστικά που εξήχθησαν. Κατασκευάστηκαν μοντέλα βασισμένα στα χαρακτηριστικά gameplay και περιεχομένου μόνο, μοντέλα από χαρακτηριστικά μέσης κίνησης κεφαλιού μόνο και μοντέλα από χαρακτηριστικά οπτικής αντίδρασης. Στη συνέχεια εξετάστηκαν μοντέλα κατασκευασμένα από το συνδυασμό διαφορετικών εισόδων από τον παίκτη.

Ακολουθεί η ανάλυση των επιλεγμένων χαρακτηριστικών και της ακριβείας των μοντέλων για κάθε σύνολο χαρακτηριστικών, καθώς και περεταίρω εξέταση των σημαντικών διαφορών μεταξύ των μοντέων που κατασκευάστηκαν για τις διάφορες κατηγορίες χαρακτηριστικών, όπως παρουσιάστηκαν στην [1].

Α. Μοντελοποίηση Εμπειρίας Παίκτη Μέσω Χαρακτηριστικών Gameplay και Περιεχομένου

Η μοντελοποίηση της εμπειρίας παίκτη από gameplay και περιεχόμενο τονίζει σημαντικές πλευρές της συμπεριφοράς του παίκτη και του σχεδιασμού του παιχνιδιού που έχουν ισχυρό αντίκτυπο στην εμπειρία gameplay. Για το λόγο αυτό, όλα τα στοιχεία του [πίνακα 1] ορίζονται ως είσοδοι για επιλογή χαρακτηριστικών και βελτιστοποίηση του μοντέλου. Τα υποσύνολα των επιλεγμένων χαρακτηριστικών, η ακρίβεια των μοντέλων και οι καλύτερες τοπολογίες MLP που βρέθηκαν ποικίλουν για τις τρεις συναισθηματικές καταστάσεις που εξετάζονται (πίνακας 3). Κατασκευάζοντας μοντέλα βασισμένα μόνο σε χαρακτηριστικά gameplay και περιεχομένου, μπορεί να προβλεφθούν οι τρεις καταστάσεις με μέση ακρίβεια (για 20 δοκιμές) υψηλότερη από 72% ενώ οι καλύτερες επιδόσεις ξεπέρασαν το 89% για προσήλωση και εκνευρισμό. Η υψηλότερη ακρίβεια για την πρόβλεψη πρόκλησης ήταν 80.6%, σημαντικά χαμηλότερη από τις αντίστοιχες για την πρόβλεψη προσήλωση και εκνευρισμού.

Αξιοσημείωτο είναι το γεγονός ότι από τα 30 διαφορετικά χαρακτηριστικά gameplay και περιεχομένου, ένα μέγιστο μόλις πέντε χαρακτηριστικών λήφθηκαν υπόψιν ως σημαντικά για την πρόβλεψη κάθε συναισθηματικής κατάστασης. Μολαταύτα, διαφορετικά υποσύνολα χαρακτηριστικών επιλέχθηκαν για κάθε συναισθηματική κατάσταση, με μόνο ένα κοινό χαρακτηριστικό μεταξύ προσήλωσης και πρόκλησης, το χρόνο που ξοδεύτηκε σε άλματα t_{jump}. Τρία από τα έξι ελεγχόμενα χαρακτηριστικά εμφανίζονται στα υποσύνολα των επιλεγμένων χαρακτηριστικών για την πρόβλεψη προσήλωσης και πρόκλησης: ο αριθμός των εχθρών, E, η τοποθέτηση των εχθρών, E_p, και ο αριθμός των powerups, N_w. Ο εκνευρισμός μπορεί να προβλεφθεί με το μικρότερο υποσύνολο χαρακτηριστικών(είχαν επιλεγεί μόνο τρία χαρακτηριστικά), όμως η ακρίβεια πρόβλεψης για τη

συγκεκριμένη συναισθηματική κατάσταση είναι σημαντικά υψηλότερη από τις αντίστοιχες για την πρόβλεψη προσήλωσης και πρόκλησης.

Αν και βρέθηκε υψηλή ακρίβεια για την πρόβλεψη των τριών συναισθηματικών καταστάσεων, η πρόκληση φαίνεται ότι είναι η ευκολότερη να μοντελοποιηθεί βάσει χαρακτηριστικών gameplay, ενώ ο εκνευρισμός η ευκολότερη.

Β. Μοντελοποίηση Εμπειρίας Παίκτη Μέσω Χαρακτηριστικών Μέσης Κίνησης Κεφαλιού

Για τη χαρτογράφηση της οπτικής συμπεριφοράς στην καταγεγραμμένη αντίδραση των παικτών, τα χαρακτηριστικά μέσης κίνησης κεφαλιού του [πίνακα 2] χρησιμοποιήθηκαν σαν είσοδοι για την επιλογή των σχετικών με την πρόβλεψη των αντιδράσεων του παίκτη χαρακτηριστικών και την βελτιστοποίηση των μοντέλων εμπειρίας παίκτη. Τα αποτελέσματα του [πίνακα 3] δείχνουν ότι τα μοντέλα που κατασκευάστηκαν από τα χαρακτηριστικά κίνησης κεφαλιού που εξήχθησαν κατά τη διάρκεια ολόκληρων συνεδριών παιχνιδιού αποδίδουν ακρίβεια ισότιμη(ίσως λίγο χαμηλότερη) με τις αντίστοιχες από τα χαρακτηριστικά gameplay.

Ανάλυση των επιλεγμένων χαρακτηριστικών δείχνει ότι η μέση οριζόντια περιστροφή του κεφαλιού ($M_{horizontal}$) είναι ένα σημαντικό χαρακτηριστικό για όλες τις συναισθηματικές καταστάσεις, ενώ οι ΟΑ και ($M_{vertical}$) βρέθηκαν χρήσιμες μόνο ως μέσα πρόβλεψης προσήλωσης και εκνευρισμού.

Το τεστ σημασίας δείχνει ότι το μοντέλο που κατασκευάστηκε για την πρόβλεψη του εκνευρισμού έχει σημαντικά υψηλότερες επιδόσεις από τα άλλα δύο μοντέλα για την πρόβλεψη προσήλωσης και πρόκλησης. Αυτό ισχύει και για τα μοντέλα που κατασκευάστηκαν βάσει χαρακτηριστικών gameplay, γεγονός που σημαίνει ότι οι τεχνικές μονής εισόδου (συμπεριφοράς ή οπτικές) είναι καλύτερες στην πρόβλεψη προσήλωσης και εκνευρισμού από ότι στην πρόβλεψη πρόκλησης.

Γ. Μοντελοποίηση Εμπειρίας Παίκτη Μέσω Χαρακτηριστικών Οπτικής Αντίδρασης

Αρχική υπόθεση της [1] ήταν ότι τα χαρακτηριστικά οπτικής αντίδρασης κατά τη διάρκεια συγκεκριμένων γεγονότων (ήττα, κρίσιμες κινήσεις κ.α.) που χρησιμοποιήθηκαν ως το μόνο κανάλι εισόδου για προσέγγιση των συναισθηματικών καταστάσεων θα απέδιδαν πιο ακριβή αποτελέσματα σε σχέση με τα χαρακτηριστικά μέσης κίνησης κεφαλιού (τα οποία αναφέρονται στη συνολική οπτική συμπεριφορά κατά τη διάρκεια του παιχνιδιού) ή με τα χαρακτηριστικά gameplay. Οι συναισθηματικές καταστάσεις φαίνεται να σχετίζονται περισσότερο με γεγνονότα που συμβαίνουν σε συγκεκριμένες στιγμές κατά τη διάρκεια του παιχνιδιού, σε αντίθεση με οπτικά χαρακτηριστικά σχετικά με ολόκληρη τη διάρκεια του παιχνιδιού. Τα χαρακτηριστικά οπτικής αντίδρασης συγχωνεύονται σε επίπεδο χαρακτηριστικών πριν την τροφοδότηση των μοντέλων πρόβλεψης και η συγχώνευση χαρακτηριστικών αναμένεται να ενισχύσει την προβλεπτική ισχύ του μοντέλου.

Η ακρίβεια πρόβλεψης εκνευρισμού λαμβάνει υψηλότερες τιμές με τη χρήση χαρακτηριστικών οπτικής αντίδρασης: η οπτική συμπεριφορά κατά τη διάρκεια αλμάτων, ήττας, τρεξίματος και αλληλεπίδρασης με διάφορα αντικείμενα φαίνεται ότι είναι αξιόπιστα μέσα πρόβλεψης εκνευρισμού. Πιο συγκεκριμένα, η παράμετρος Εκφραστικότητας Ενέργειας κατά την αλληλεπίδραση με αντικείμενα (Po_{item}) και την εκκίνηση τρεξίματος ($Po_{startRun}$), καθώς και η Ολική Ενεργοποίηση κατά την ήττα (Oa_{lose}) συνδέονται πολύ συχνά με τον εκνευρισμό. Εκτός από τον εκνευρισμό, πολύ καλή ακρίβεια επιτεύχθη και για τη χρήση χαρακτηριστικών οπτικής αντίδρασης για την πρόβλεψη πρόκλησης, με τόσο τον εκνευρισμό όσο και την πρόκληση να ξεπερνούν την ακρίβεια πρόβλεψης για την προσήλωση.

Δ. Συγχώνευση Χαρακτηριστικών για τη Μοντελοποίηση Εμπειρίας Παίκτη

Αυτή η υποπαράγραφος παρουσιάζει πειράματα με δικόρυφα* χαρακτηριστικά ως εισόδους στα προβλεπτικά μοντέλα. Πρώτα συγχωνεύονται τα χαρακτηριστικά gameplay/περιεχομένου και μέσης κίνησης κεφαλιού και στη συνέχεια εξετάζεται το αντίκτυπο της συγχώνευσης μεταξύ χαρακτηριστικών gameplay/περιεχομένου και οπτικής αντίδρασης στην ακρίβεια πρόβλεψης των μοντέλων.

	One Modality		Bimodality		
	Gameplay/Content	Mean Head Movement	Visual Reaction	Gameplay/Mean Head Movement	Gameplay/Visual Reaction
			Engagement		
Selected features	t_{jump} k_{stomp} $\boldsymbol{N_w}$ t_{run} $\boldsymbol{E_p}$	OA Avg $M_{vertical}$ $M_{horizontal}$ SE PO	OA_{endRun} FL_{endRun} FL_{stomp} $PO_{startLeft}$ PO_{move} $TE_{endRight}$ $Avg_{endJump}$	t_{jump} k_{stomp} $\boldsymbol{N_w}$ $M_{horizontal}$ t_{comp} n_{boxes} $M_{vertical}$ FL	$TE_{endRight}$ t_{jump} $\boldsymbol{B}$ PO_{stomp} $TE_{endJump}$
ANN topology	6	4	$4-6$	4-6	2
P	78.69%	74.23%	78.06%	77.78%	**83.97%**
P_{max}	89.68%	78.57%	86.51%	89.68%	**91.27%**
			Frustration		
Selected features	$t_{lastLife}$ n_{boxes} t_{left}	$M_{horizontal}$ OA TE FL $M_{vertical}$	OA_{lose} Avg_{stomp} SE_{endRun} $PO_{startRun}$ $M_{endJump}$ PO_{item}	$t_{lastLife}$ TE n_{gJump} n_{boxes} PO d_{total} OA	FL_{item} FL_{stomp} t_{small} FL_{lose} n_{boxes}
ANN topology	$8-2$	4	$8-10$	$4-4$	8
P	83.5%	83.04%	**86.21%**	83.71%	85.92%
P_{max}	89.17%	89.17%	**92.5%**	92.5%	89.17%
			Challenge		
Selected features	t_{jump} $\boldsymbol{E}$ $k_{unleashed}$ d_{total}	$M_{horizontal}$ FL PO	FL_{lose} $PO_{startRun}$ $FL_{startRun}$ $Avg_{endLeft}$	$M_{hotizontal}$ t_{jump} t_{run} t_{big} $k_{unleashed}$ t_{small}	FL_{item} $FL_{startRun}$ $M_{startJump}$ FL_{lose} $SE_{endLeft}$ t_{left} $Avg_{startJump}$
ANN topology	$4-2$	$4-8$	$10-8$	$10-10$	$10-10$
P	72.36%	75%	**84.13%**	77.36%	78.40%
P_{max}	80.56%	79.17%	**88.88%**	85.41%	86.81%

1) Μοντελοποίηση Μέσω Χαρακτηριστικών Gameplay/Περιεχομένου και Μέσης Κίνησης Κεφαλιού: Η χρήση χαρακτηριστικών κίνησης κεφαλιού για ολόκληρη τη διάρκεια των συνεδριών παιχνιδιού και χαρακτηριστικών gameplay/περιεχομένου αποδίδει ακριβή αποτελέσματα για την πρόβλεψη προσήλωσης, εκνευρισμού και πρόκλησης. Επιλέχθηκαν διαφορετικά χαρακτηριστικά gameplay και κίνησης κεφαλιού για κάθε συναισθηματική κατάσταση. Η μέση οριζόντια και κάθετη κατευθυντικότητα του κεφαλιού, μαζί με τη ρευστότητα στην κίνηση και τα χαρακτηριστικά gameplay/περιεχομένου (ο αριθμός των εχθρών που σκοτώθηκαν από άλματα, ο χρόνος που ξοδεύτηκε τρέχοντας και ολοκληρώνοντας το παιχνίδι και τα powerups) είχαν ως αποτέλεσμα μοντέλο πρόβλεψης προσήλωσης με μέγιστη ακρίβεια 89.68%. Κάποια από αυτά τα χαρακτηριστικά (όπως ο αριθμός των powerups, N_w, ο χρόνος που ξοδεύτηκε σε άλματα, t_{jump}, η μέση οριζόντια και κάθετη κατεύθυνση του κεφαλιού, $M_{horizontal}$ και $M_{vertical}$) εμφανίζονται και στο υποσύνολο χαρακτηριστικών που επιλέχθηκε κατά την κατασκευή μοντέλων για κάθε μια από αυτές τις τεχνικές μεμονωμένα. Αυτό αποδεικνύει τη

σημασία των χαρακτηριστικών ως μέσων πρόβλεψης της προσήλωσης του παίκτη.

Το επιλεγμένο υποσύνολο χαρακτηριστικών για την πρόβλεψη του εκνευρισμού περιλαμβάνει: χρονική (ΤΑ), ενέργεια και παραμέτρους εκφραστικότητας ΟΑ καθώς και τις $t_{lastLife}$, n_{gJump}, n_{boxes} και d_{total}. Τα χαρακτηριστικά ΤΑ και ΟΑ εμφανίζονται επίσης και στο υποσύνολο χαρακτηριστικών που επιλέχθηκε για την πρόβλεψη του εκνευρισμού μέσω χρήσης χαρακτηριστικών μέσης κίνησης κεφαλιού μόνο. Όπως ήταν αναμενόμενο, ο χρόνος που ξοδεύτηκε παίζοντας την τελευταία ζωή ($t_{lastLife}$) και ο αριθμός των κουτιών που χτυπήθηκαν ή καταστράφηκαν (n_{boxes}) είναι σημαντικοί παράγοντες πρόβλεψης του εκνευρισμού. Αυτά τα χαρακτηριστικά του gameplay εμφανίζονται και στα μοντέλα που κατασκευάστηκαν βάσει μόνο των χαρακτηριστικών gameplay.

Τα χαρακτηριστικά που επιλέχθηκαν για την πρόβλεψη της πρόκλησης είναι κυρίως χαρακτηριστικά gameplay σχετικά με το χρόνο, τα οποία συγχωνεύονται με την μέση οριζόντια περιστροφή κεφαλιού ($M_{horizontal}$). Τα χαρακτηριστικά gameplay που επιλέχθηκαν και εμφανίζονται επίσης στο υποσύνολο χαρακτηριστικών επιλεγμένων για την πρόβλεψη της πρόκλησης με μόνη είσοδο το gameplay, περιλαμβάνουν τον χρόνο αλμάτων (t_{jump}) και τον αριθμό των εχθρών που σκοτώθηκαν εξαπολύοντας ένα καβούκι χελώνας ($k_{uleashed}$). Τα νέα σχετικά με το χρόνο χαρακτηριστικά gameplay που επιλέχθηκαν (t_{run}, t_{big} και t_{small}) έχουν ως αποτέλεσμα την αύξηση κατά 5% της μέσης απόδοσης (σε σχέση με τη μέση απόδοση των μοντέλων βασισμένων μόνο στα χαρακτηριστικά gameplay), υποδεικνύοντας τη σημασία του χρόνου τρεξίματος και του χρόνου για τον οποίο ο Μάριο είχε τη μικρή ή τη μεγάλη μορφή ως παράγοντες πρόβλεψης της πρόκλησης για τον παίκτη.

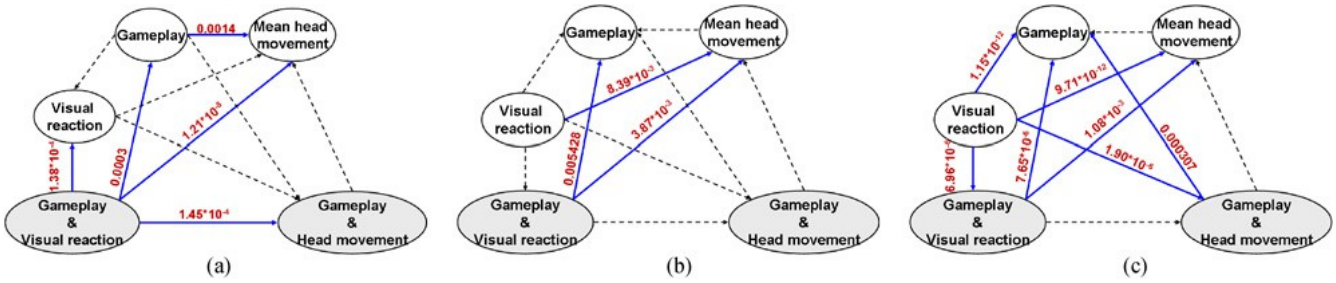
Η ακρίβεια του μοντέλου που κατασκευάστηκε για την πρόβλεψη του εκνευρισμού αποδεικνύεται σημαντικά υψηλότερη από τις αντίστοιχες για την πρόβλεψη προσήλωσης και πρόκλησης (το πόρισμα αυτό είναι παρόμοιο με τις παρατηρήσεις για τη δοκιμή των διαφορών στις μέσες τιμές απόδοσης μεταξύ μοντέλων κατασκευασμένων αποκλειστικά βάσει χαρακτηριστικών gameplay και μοντέλων κατασκευασμένων αποκλειστικά βάσει χαρακτηριστικών μέσης κίνησης κεφαλιού.

2) Μοντελοποίηση Μέσω Χαρακτηριστικών Gameplay/Περιεχομένου και Οπικής Αντίδρασης: Ο συνδυασμός χαρακτηριστικών gameplay/περιεχομένου και οπτικής αντίδρασης, έχει ως αποτέλεσμα την εμφάνιση χαρακτηριστικών που δεν χρησιμοποιήθηκαν όταν εξετάσθηκε η κάθε τεχνική ξεχωριστά. Αυτό μπορεί να αποδοθεί στο γεγονός ότι υπάρχουν σχέσεις μεταξύ χαρακτηριστικών που χρησιμοποιούνται από το gameplay/περιεχόμενο και χαρακτηριστικών οπτικής αντίδρασης μεμονωμένα. Η επιλογή των χαρακτηριστικών κινείται πέραν των γραμμικά συσχετισμένων χαρακτηριστικών, επομένως νέα επιλεγμένα υποσύνολα χαρακτηριστικών είναι αναμενόμενο να προκύψουν με σκοπό τη μεγιστοποίηση ακριβείας της απόδοσης.
Για την προσήλωση, ένα μικρότερο υποσύνολο συνδυασμένων χαρακτηριστικών είχε ως αποτέλεσμα υψηλότερη ακρίβεια από τη χρήση μεγαλυτέρων συνόλων χαρακτηριστικών από κάθε μία από τις δύο τεχνικές εισόδου ξεχωριστά. Τα περισσότερα από τα επιλεγμένα χαρακτηριστικά δεν εμφανίζονται στο υποσύνολο που επιλέχθηκε για την πρόβλεψη προσήλωσης σε κάθε μια από τις τεχνικές αυτές. Η πλειοψηφία των επιλεγμένων χαρακτηριστικών είναι άμεσα ή έμμεσα συνδεδεμένη με την κίνηση του κεφαλιού και τα συμβάντα gameplay που αφορούν τα άλματα. Η t_{jump} είναι μια ένδειξη του χρόνου που αφιερώνεται σε άλματα, PO_{stomp} είναι η ενέργεια της κίνησης του κεφαλιού όταν ο Μάριο πηδάει στο κεφάλι ενός εχθρού και τον σκοτώνει, $TE_{endJump}$ είναι η παράμετρος χρονικής εκφραστικότητας κατά την προσγείωση και η Β αναφέρεται στον αριθμό των κουτιών, η αλληλεπίδραση με τα οποία απαιτεί ένα άλμα. Φαίνεται επομένως ότι η στιγμή του άλματος αποτελεί παράγοντα για την πρόβλεψη της προσήλωσης σε παιχνίδια πλατφόρμας, καθώς η μέση ακρίβεια για προσήλωση (83.97%) μέσω της διπλής συγχώνευσης χαρακτηριστικών gameplay

και οπτικής αντίδρασης είναι η βέλτιστη σε σχέση με οποιονδήποτε άλλο τύπο χαρακτηριστικών ως είσοδο μοντέλου.

Το επιλεγμένο υποσύνολο χαρακτηριστικών για την πρόβλεψη του εκνευρισμού περίεχει λιγότερα χαρακτηριστικά από τις επιλογές που έγιναν χωριστά για κάθε τεχνική. Ενδιαφέρον παρουσιάζει το γεγονός ότι δεν υπάρχει επικάλυψη μεταξύ των χαρακτηριστικών που επιλέχθηκαν από τα συγχωνευμένα χαρακτηριστικά και αυτά που επιλέχθηκαν από τα χαρακτηριστικά οπτικής αντίδρασης, ενώ υπάρχει μόνο ένα κοινό χαρακτηριστικό (n_{boxes}) μεταξύ των επιλεγμένων συγχωνευμένων χαρακτηριστικών και αυτών που επιλέχθηκαν από το gameplay.

Το επιλεγμένο υποσύνολο χαρακτηριστικών για την πρόβλεψη της πρόκλησης περιέχει μεγάλο αριθμό χαρακτηριστικών σε σύγκριση με αυτά που επιλέχθηκαν για κάθε τεχνική μεμονωμένα. Κοιτάζοντας τα χαρακτηριστικά που επιλέχθηκαν για τις τρεις μεθόδους—τα μοντέλα κατασκευασμένα από χαρακτηριστικά gameplay, το μοντέλο κατασκευασμένο από χαρακτηριστικά οπτικής αντίδρασης και το μοντέλο κατασκευασμένο από τη συγχώνευση των δύο αυτών τεχνικών—φαίνεται ότι υπάρχουν δύο επικαλύψεις μεταξύ των επιλεγμένων χαρακτηριστικών οπτικής αντίδρασης ($FL_{startRun}$ και FL_{lose}) και κανένα κοινό χαρακτηριστικό gameplay. Η μέση απόδοση που προκύπτει για την πρόκληση (78.4%) υποδηλώνει ότι τα νέα επιλεγμένα χαρακτηριστικά δεν βελτιώνουν την προβλεπτική ισχύ του μοντέλου σε σχέση με την αντίστοιχη απόδοση των χαρακτηριστικών οπτικής αντίδρασης. Η στατιστική ανάλυση δε δείχνει κάποια σημαντική διαφορά απόδοσης μεταξύ των μοντέλων που κατασκευάστηκαν για πρόβλεψη προσήλωσης και εκνευρισμού, ενώ οι αποδόσεις των δύο αυτών μοντέλων είναι σημαντικά υψηλότερες από την απόδοση του μοντέλου που κατασκευάστηκε για την πρόβλεψη της πρόκλησης.



Ε. Στατιστική Ανάλυση

Παρουσιάζεται η στατιστική ανάλυση που έγινε στην [1] για τον εντοπισμό πιθανών σημαντικών διαφορών στην ακρίβεια των μοντέλων που κατασκευάστηκαν για τις διάφορες κατηγορίες χαρακτηριστικών. Η Εικ. 6(^^) παρουσιάζει τα αποτελέσματα των δοκιμών για σημαντικές διαφορές στην απόδοση μεταξύ των μοντέλων που κατασκευάστηκαν για τις τρεις συναισθηματικές καταστάσεις σε όλες τις κατηγορίες χαρακτηριστικών. Τα συνεχή βέλη απεικονίζουν σημαντικές διαφορές στη μέση απόδοση, ενώ τα διακεκομμένα βέλη απεικονίζουν ασήμαντες στατιστικά διαφορές στη μέση απόδοση. Παρουσιάζονται επίσης οι τιμές-p* που βρέθηκαν από τις στατιστικά σημαντικές διαφορές.

Όπως φαίνεται στην εικ. 6, τα χαρακτηριστικά μέσης κίνησης κεφαλιού δεν φέρουν υψηλή απόδοση σε σχέση με τα υπόλοιπα χαρακτηριστικά όταν χρησιμοποιούνται μόνα τους. Όλα τα μοντέλα που κατασκευάστηκαν από άλλα σύνολα χαρακτηριστικών αποδίδουν καλύτερα έως και σημαντικά καλύτερα από το μοντέλο που κατασκευάστηκε βάσει των χαρακτηριστικών μέσης κίνησης κεφαλιού για την προσήλωση. Αυτά τα χαρακτηριστικά ωστόσο, αποδίδουν καλύτερα (χωρίς σημαντική διαφορά) από τα μοντέλα κατασκευασμένα από χαρακτηριστικά gameplay για την πρόβλεψη

εκνευρισμού και πρόκλησης. Συγχωνεύοντας τα χαρακτηριστικά μέσης κίνησης κεφαλιού με τα χαρακτηριστικά gameplay παρ' όλα αυτά, επιτεύχθηκε μεγαλύτερη ακρίβεια από ότι όταν χρησιμοποιήθηκαν μόνο χαρακτηριστικά μέσης κίνησης κεφαλιού για την κατασκευή μοντέλων εμπειρίας παίκτη για όλες τις συναισθηματικές καταστάσεις. Τα ποσοστά ακριβείας που επιτεύχθηκαν είναι ακόμα καλύτερα από τα αντίστοιχα των χαρακτηριστικών gameplay για την πρόβλεψη εκνευρισμού και πρόκλησης.

Τα αποτελέσματα των μοντέλων που κατασκευάστηκαν βάσει χαρακτηριστικών οπτικής αντίδρασης από την άλλη, είναι καλύτερα από αυτά των μοντέλων που κατασκευάστηκαν βάσει χαρακτηριστικών μέσης κίνησης κεφαλιού ή gameplay για την πρόβλεψη εκνευρισμού και πρόκλησης. Τα μοντέλα αυτά βελτιώνουν και αυτά που κατασκευάστηκαν βάσει συγχωνευμένων χαρακτηριστικών gameplay και μέσης κίνησης κεφαλιού για όλες τις συναισθηματικές καταστάσεις.

Συγχωνεύοντας χαρακτηριστικά οπτικής αντίδρασης και χαρακτηριστικά gameplay, κατασκευάστηκαν μοντέλα με υψηλότερη απόδοση στην πρόβλεψη προσήλωσης σε σχέση με οποιοδήποτε άλλο μοντέλο που κατασκευάστηκε με βάση οποιοδήποτε άλλο σύνολο χαρακτηριστικών. Το επιχείρημα αυτό ισχύει επίσης για τον εκνευρισμό και την πρόκληση, με εξαίρεση το μοντέλο που κατασκευάστηκε από χαρακτηριστικά οπτικής αντίδρασης, το οποίο είχε υψηλότερη απόδοση από το μοντέλο συγχώνευσης αυτών με χαρακτηριστικά gameplay.

Η συγχώνευση χαρακτηριστικών από διαφορετικές τεχνικές, φαίνεται γενικότερα να έχει ως αποτέλεσμα πιο ακριβή μοντέλα πρόβλεψης των αντιδράσεων των παικτών σε σχέση με την κατασκευή μοντέλων βάσει χαρακτηριστικών μιας μόνο κατηγορίας. Η συγχώνευση των χαρακτηριστικών ενισχύει τα μοντέλα με περισσότερη γνώση για περισσότερα από ένα κανάλια πληροφοριών, το οποίο φαίνεται να έχει θετικό αντίκτυπο στην απόδοση των μοντέλων.

Ήταν αναμενόμενο ότι η συγχώνευση χαρακτηριστικών gameplay και οπτικής αντίδρασης θα απέφερε υψηλότερα ποσοστά ακριβείας από τη χρήση οποιουδήποτε άλλου συνόλου χαρακτηριστικών, αλλά η υπόθεση αυτή δεν ισχύει για την κατάσταση της πρόκλησης. Μολαταύτα, τα μοντέλα που κατασκευάστηκαν για την πρόβλεψη πρόκλησης από χαρακτηριστικά οπτικής αντίδρασης και τη συγχώνευση τους με χαρακτηριστικά gameplay είναι πολυεπίπεδα perceptrons δύο κρυμμένων επιπέδων, γενονός που συνεπάγεται ότι η σχέση μεταξύ των επιλεγμένων χαρακτηριστικών και των καταγεγραμμένων προτιμήσεων των παικτών είναι πιο πολύπλοκη από απλούς γραμμικούς συσχετισμούς.

Η ελάττωση της απόδοσης όταν τα χαρακτηριστικά συγχωνεύονται θεωρείται αποτέλεσμα του τρόπου επιλογής των χαρακτηριστικών, ο οποίος αποτυγχάνει να επιλέξει το βέλτιστο υποσύνολο χαρακτηριστικών για πρόβλεψη όταν η ποσότητα των χαρακτηριστικών από τα οποία επιλέγουμε γίνεται μεγάλη. Για παράδειγμα, όταν συγχωνεύονται χαρακτηριστικά gameplay με χαρακτηριστικά οπτικής αντίδρασης, προκύπτει ένα τελικό σύνολο 114 χαρακτηριστικών.

Factors	Engagement	Frustration	Challenge
(A)	**0.00001**	0.78	**0.03**
(B)	$\mathbf{4.21 * 10^{-6}}$	**0.0004**	$\mathbf{3.54 * 10^{-9}}$
(A*B) Interaction	0.13	0.512	$\mathbf{1.05 * 10^{-6}}$

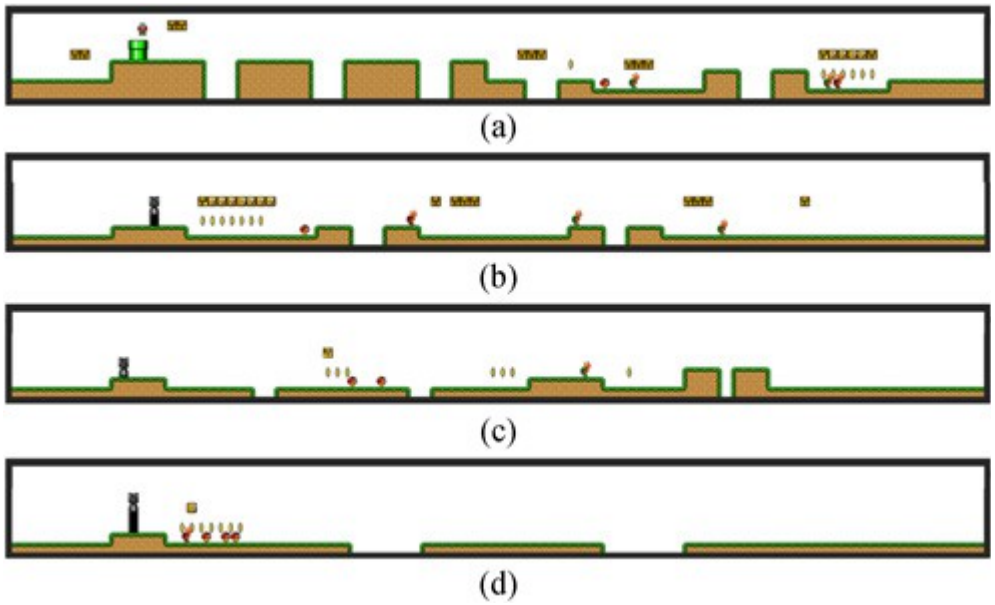
Για περεταίρω ανάλυση της επίδρασης που έχει η αλληλεπίδραση μεταξύ των χαρακτηριστικών στα ποσοστά ακριβείας των μοντέλων, εφαρμόστηκε μια αμφίδρομη δοκιμή ANOVA. Για το πείραμα αυτό, υπολογίζονται δύο παράγοντες: 1) η ύπαρξη (αντί της μη ύπαρξης) των χαρακτηριστικών gameplay για την πρόβλεψη αντιδράσεων, και 2) η ύπαρξη χαρακτηριστικών οπτικής αντίδρασης (αντί χαρακτηριστικών κίνησης κεφαλιού). Με αυτόν τον τρόπο γίνεται ευκολότερη η εξέταση του αν η χρήση χαρακτηριστικών οπτικής αντίδρασης (ή εναλλακτικά κίνησης κεφαλιού) ή η συγχώνευση στοιχείων gameplay με οπτικά στοιχεία θα απέφερε σημαντικές αλλαγές στην απόδοση των μοντέλων.

Τα αποτελέσματα μιας 2 x 2 [(gameplay και όχι-gameplay) x (οπτική αντίδραση και κίνηση κεφαλιού)] αμφίδρομης μεταξύ-ομάδων ANOVA φαίνονται στον πίνακα 4.

Και οι δύο ανεξάρτητες μεταβλητές φαίνεται να έχουν αντίκτυπο στην πρόβλεψη προσήλωσης με τιμές-p* 0.0001 kai $4.21*10^{-6}$, αντίστοιχα. Παρ' όλα αυτά, δεν ανιχνεύθηκε σημαντική επίδραση κατά την ανάλυση της αλληλεπίδρασης μεταξύ των μεταβλητών (p-value=0.13). Όσον αφορά τον εκνευρισμό, τα αποτελέσματα έδειξαν σημαντικές διαφορές μόνο για το δεύτερο παράγοντα (p-value=0.0004) ενώ δεν εντοπίστηκαν σημαντικές επιδράσεις για τον πρώτο παράγοντα (p-value=0.78) ή για την αλληλεπίδραση μεταξύ των παραγόντων (p-value=0.512). Τέλος, για την πρόκληση σημαντικά αποτελέσματα είχαμε και για τους δύο παράγοντες(p-value=0.03 και p-value=$3.54*10^{9}$) και για την αλληλεπίδραση μεταξύ τους (p-value=$1.05*10^{-6}$). Τα αποτελέσματα αυτά υποδηλώνουν ότι ο τύπος των οπτικών ερεθισμάτων είχε σημαντικό αντίκτυπο στην ακρίβεια πρόβλεψης για τις τρεις συναισθηματικές καταστάσεις, ενώ η συμπερίληψη των χαρακτηριστικών gameplay αποδείχθηκε ότι είχε σημαντική επίδραση στην πρόβλεψη προσήλωσης και πρόκλησης. Η αλληλεπίδραση μεταξύ χαρακτηριστικών gameplay και οπτικής αντίδρασης από την άλλη πλευρά, είχε σημαντική επίδραση μόνο στην πρόβλεψη της πρόκλησης.

1.1.6 Χρήση Μοντέλων Εμπειρίας Παίκτη για Παραγωγή Εξατομικευμένων Επιπέδων

Ο απώτερος στόχος για την κατασκευή μοντέλων εμπειρίας παίκτη που βασίζονται σε δεδομένα είναι αυτά τα μοντέλα να χρησιμοποιηθούν για να κλείσουν τον συναισθηματικό βρόχο [2],[8],[9] στο παιχνίδι προσαρμόζοντας τη δημιουργία υλικού του παιχνιδιού στις ανάγκες και το χαρακήρα κάθε διαφορετικού παίκτη. Στα πειράματα που παρουσιάζονται σε αυτό το μέρος, περιγράφεται η μέθοδος που ακολουθείται για την προσαρμογή της δημιουργίας υλικού, βασισμένη στα μοντέλα εμπειρίας παίκτη που κατασκευάστηκαν στο προηγούμενο κομμάτι. Συγκενρωνόμαστε στα μοντέλα που χτίστηκαν πάνω σε επιλεγμένα χαρακτηριστικά από gameplay και οπτική αντίδραση καθώς τα μοντέλα αυτά δίνουν τη μεγαλύτερη ακρίβεια για την πρόβλεψη προσήλωσης και υψηλή ακρίβεια με πληθώρα πληροφοριών για τη συμπεριφορά του παίκτη και τα οπτικά ερεθίσματα κατά την πρόβλεψη εκνευρισμού και πρόκλησης.



Εικ 7: Παραδείγματα επιπέδων δημιουργημένων για τη μεγιστοποίηση της προβεπόμενης πρόκλησης και του προβλεπόμενου εκνευρισμού για 2 παίκτες με διαφορετικά χαρακτηριστικά οπτικής αντίδρασης. (a) Επίπεδο δημιουργημένο για μέγιστο εκνευρισμό (Παίκτης 1). (b) Επίπεδο δημιουργημένο για μέγιστο εκνευρισμό (Παίκτης 2). (c) Επίπεδο δημιουργημένο για μέγιστη πρόκληση (Παίκτης 1). (d) Επίπεδο δημιουργημένο για μέγιστη πρόκληση (Παίκτης 2).

Τα μοντέλα εμπειρίας παίκτη που κατασκευάστηκαν χρησιμοποιούνται για να προσαρμόσουν το περιεχόμενο του παιχνιδιού σε μεμονωμένους παίκτες. Σαν πρώτο βήμα προς αυτή τη διαδικασία υιοθετήθηκε η μεθοδολογία που προτείνεται στο [5] για κατασκευή μοντέλων που επιτρέπουν έλεγχο του περιεχομένου επιβάλλοντας ελεγχόμενα χαρακτηριστικά στην είσοδο των ANNs. Στη συνέχεια, με σκοπό τη δημιουργία επιπέδων προσαρμοσμένων σε κάθε παίκτη, διερευνάται εξοντωτικά ο χώρος του περιεχομένου με σκοπό την εύρεση ενός συνδυασμού τιμών για τα χαρακτηριστικά περιεχομένου που αποφέρει (μαζί με τα επιλεγμένα χαρακτηριστικά gameplay και οπτικής αντίδρασης) την μέγιστη τιμή εξόδου της ANN για την υπό εξέταση συναισθηματική κατάσταση (προσήλωση, πρόκληση και εκνευρισμός). Οι λεπτομέρειες αυτής της προσέγγισης μπορούν να βρεθούν σε παλιότερη δουλειά [5] των συγγραφέων της συγγενικής [1]. Ενδεικτικά για τα μοντέλα εμπειρίας παίκτη που κτίστηκαν, ο χώρος αναζήτησης αποτελείται από ένα μέγιστο από πέντε χαρακτηριστικά περιεχομένου: τον αριθμό των κενών, το μέσο πλάτος των κενών, τον αριθμό των εχθρών, την τοποθέτηση των εχθρών και τον αριθμό των κουτιών με εμβέλειες τιμών [2, 6], [5, 15], [3, 7], [0, 2] και [0, 15] αντίστοιχα. Ο χώρος αναζήτησης εξερευνάται ξεκινώντας από τις ελάχιστες δυνατές τιμές και σε κάθε βήμα οι τιμές αυξάνονται κατά ένα. Με τόσο μικρό χώρο αναζήτησης (13200 σχηματισμοί) μπορούμε να βρούμε το βέλτιστο σχηματισμό σχεδόν στιγμιαία, επιτρέποντας τη δημιουργία επιπέδων σε πραγματικό χρόνο.

Σαν πείραμα απόδειξης της έννοιας, δημιουργούμε επίπεδα που μεγιστοποιούν τον προβλεπόμενο εκνευρισμό και την προβλεπόμενη πρόκληση για δύο παίκτες με διαφορετικά χαρακτηριστικά οπτικής αντίδρασης που δεν χρησιμοποιήθηκαν για κατασκευή μοντέλων. Χρησιμοποιώντας τον βασισμένο στην εμπειρία PCG μηχανισμό που προτείνεται στο [5], δημιουργήθηκε ένα νέο επίπεδο για κάθε παίκτη που βελτιστοποιεί τις δύο αυτές καταστάσεις

προβεπόμενης εμπειρίας παίκτη (Εικ 7). Είναι προφανές ότι ο οδηγημένος από την εμπειρία PCG μηχανισμός δημιουργεί μια ποικιλία από εξατομικευμένα επίπεδα, ανάλογα με τα οπτικά ερεθίσματα και τα στοιχεία συμπεριφοράς του παίκτη. Παραδείγματος χάριν, φαίνεται ότι ένα επίπεδο μπορεί να είναι πιο εκνευριστικό για τον πρώτο παίκτη όταν περιέχει περισσότερα κενά με μικρό πλάτος, μεγάλο αριθμό κουτιών και εχθρούς διασκορπισμένους τυχαία. Ένα επίπεδο με λιγότερα κενά με μικρό πλάτος και εχθρούς γύρω τους είναι πιο εκνευριστικό για το δεύτερο παίκτη. Παρομοίως, ένα προκλητικό επίπεδο για τον πρώτο παίκτη είναι εκείνο που περιέχει κενά μικρού πλάτους, μικρό αριθμό εχθρών διασκορπισμένων τυχαία στο επίπεδο και καθόλου κουτιά. Ένα επίπεδο με ελαφρά πιο προκλητικές πτυχές έχει δημιουργηθεί για το δεύτερο παίκτη, με μικρότερο αριθμό κενών αλλά με μεγαλύτερο πλάτος και εχθρούς τοποθετημένους γύρω από συλλεγόμενα αντικείμενα.

Τα δεδομένα συμπεριφοράς και η εμπειρία που ανέφερε ο κάθε παίκτης για τα δημιουργημένα επίπεδα δεν είναι διαθέσιμα, επομένως δεν υπάρχει εγγύηση ότι ο μηχανισμός προσαρμογής δημιουργεί υψηλότερα επίπεδα πρόκλησης και εκνευρισμού. Ωστόσο, τα μοντέλα υψηλής ακριβείας ANN (πάνω από 80% ακριβή)--που καθοδηγούν τη δημιουργία των επιπέδων--υποδηλώνουν ότι κατά πάσα πιθανότητα επιτυγχάνονται υψηλότερες τιμές για όλες τις συναισθηματικές καταστάσεις. Επιπροσθέτως, μια προηγούμενη μελέτη χρηστών του Super Mario Bros [5]--όπου ακολουθήθηκε η ίδια προσέγγιση εξοντωτικής αναζήτησης για τη δημιουργία εξατομικευμένων επιπέδων με βάση απλούστερα μοντέλα—απέδειξε ότι τα εξατομικευμένα επίπεδα προτιμούνται από την πλειοψηφία των παικτών.

1.1.7 Συμπεράσματα

Παρουσιάστηκε ένα εκτενές σύνολο πειραμάτων για τη μοντελοποίηση της εμπειρίας παίκτη σε βιντεοπαιχνίδια, βασισμένο σε δύο τρόπους εισόδου από τον παίκτη: συμπεριφορικά δεδομένα από το gameplay και την οπτική συμπεριφορά του παίκτη. Έγινε συλλογή ενός μεγάλου σώματος δεδομένων οπτικής συμπεριφοράς και εμπειρίας παίκτη από 58 παίκτες Super Mario Bros και κατασκευάστηκαν χρησιμοποιώντας μηχανισμό σύζευξης αυτόματης επιλογής χαρακτηριστικών και νευρο-εξελισσόμενης εκμάθησης προτιμήσεων. Είδαμε ότι οι οπτικές αντιδράσεις των παικτών σε συγκεκριμένα συμβάντα στο παιχνίδι μπορούν να αποτελέσουν πλούσια πηγή πληροφοριών σχετικών με τις προτιμήσεις τους όσον αφορά την πρόκληση και τον εκνευρισμό (φτάνοντας ποσοστά ακριβείας μοντέλου 88.88% και 92.5%, αντίστοιχα). Ωστόσο, η προσήλωση (η υψηλότερη ακρίβεια μοντέλου που επιτεύχθηκε ήταν 91.27%) φαίνεται ότι είναι μια έννοια συνδεδεμένη τόσο με τον τρόπο που ένα παιχνίδι έχει σχεδιαστεί, παίζεται, τόσο και με τις οπτικές πληροφορίες που προέρχονται από τον ίδιο τον παίκτη.

Περεταίρω μελλοντική δουλειά θα μπορούσε να περιλαμβάνει δοκιμές για τη γενικότερη εφαρμογή της προτεινόμενης μεθοδολογίας και τα αποτελέσματα της. Ενώ το Super Mario Bros είναι ουσιαστικά ο ορισμός του είδους των παιχνιδιών πλατφόρμας, θα είχε μεγάλο ενδιαφέρον η εξέταση του κατά πόσο η προτεινόμενη μεθοδολογία μπορεί να γενικευτεί σε άλλα είδη βιντεοπαιχνιδιών όπως τα shooters πρώτου προσώπου (FPS) ή σοβαρά παιχνίδια. Η [1] υποστηρίζει ότι η προσέγγιση που παρουσιάστηκε έχει τη δυνατότητα να εφαρμοστεί επιτυχημένα σε τέτοια παιχνίδια, καθώς τα περισσότερα από τα χαρακτηριστικά gameplay που ορίστηκαν μπορούν εύκολα να γενικευτούν για την σύλληψη τρόπων παιξίματος για μια μεγάλη ποικιλία άλλων παιχνιδιών. Η εφαρμογή των χαρακτηριστικών οπτικής αντίδρασης (τα οποία αποδείχθηκαν αποτελεσματικά μέσα πρόβλεψης της κατάστασης του παίκτη) φαίνεται να είναι μια τετριμμένη διαδικασία καθώς η εξαγωγή αυτών εξαρτάται από σημαντικά συμβάντα σε γενικό πλαίσιο (όπως δείκτες ήττας ή νίκης).

Κάποιοι περιορισμοί είναι έμφυτοι στην προσέγγιση μοντελοποίησης εμπειρίας παίκτη που ακολουθήθηκε. Η μέθοδος επιλογής χαρακτηριστικών παρέχει έναν αποτελεσματικό μηχανισμό για την επιλογή σχετικών χαρακτηριστικών όταν το μέγεθος του χώρου αναζήτησης είναι πολύ μικρό.

Αυτή η μέθοδος ωστόσο, οδηγεί σε αναντίστοιχο υποσύνολο χαρακτηριστικών κατά την αναζήτηση σε μεγάλο χώρο. Η αυτόματη επιλογή χαρακτηριστικών είναι ένα απαραίτητο βήμα κατά την κατασκευή των μοντέλων εμπειρίας διότι η επιλογή του σωστού υποσυνόλου χαρακτηριστικών μπορεί να έχει μεγάλο αντίκτυπο στην ακρίβεια της πρόβλεψης που γίνεται. Ένας τρόπος βελτίωσης της ακριβείας πρόβλεψης των μοντέλων είναι η βελτίωση των δυνατοτήτων καθολικής αναζήτησης της διαδικασίας επιλογής χαρακτηριστικών. Αλγόριθμοι που βασίζονται σε μετα-ευριστική αναζήτηση, όπως η γενετική αναζήτηση χαρακτηριστικών [10] μπορεί να βελτιώσει τον εντοπισμό καταλληλότερων υποσυνόλων χαρακτηριστικών.

Ένας άλλος περιορισμός της προτεινόμενης μεθόδου μοντελοποίησης αφορά την εκφραστικότητα των μοντέλων εμπειρίας παίκτη. Χρησιμοποιώντας νευρο-εξελισσόμενη εκμάθηση προτιμήσεων, έχουμε το πλεονέκτημα της ικανότητας καθολικής προσέγγισης για την κατασκευή μη γραμμικών μοντέλων μεγάλης ακριβείας αλλά χάνουμε τη δυνατότητα εύκολης ανάλυσης των σχέσεων αιτίου-αποτελέσματος μεταξύ των επιλεγμένων χαρακτηριστικών και της πρόβλεψης των μοντέλων για κάθε συναισθηματική κατάσταση. Με αυτόν τον τρόπο η εκμετάλλευση της χρήσης πιο εκφραστικών μοντέλων, όπως τα δέντρα αποφάσεων ή τα ασαφή νευρωνικά δίκτυα, για την μοντελοποίηση της εμπερίας του παίκτη αποτελεί μια μελλοντική κατεύθυνση.

Όπως αποδείχθηκε με πείραμα απόδειξης της έννοιας στην [1], ένας σχεδιαστής επιπέδου μπορεί να χρησιμοποιήσει αυτά τα μοντέλα εμπειρίας παίκτη και να δημιουργήσει αυτόματα εξατομικευμένα επίπεδα για κάθε παίκτη. Δεδομένου συνόλου χαρακτηριστικών συμπεριφοράς και οπτικής αντίδρασης ενός παίκτη, τα μοντέλα εμπειρίας παίκτη ANN μπορούν να ενημερώσουν το σχεδιαστή σχετικά με το σύνολο χαρακτηριστικών επιπέδου του παιχνιδιού (όπως οι αριθμοί εχθρών και κενών) που μπορούν να μεγιστοποιήσουν (ή ακόμα και να ελαχιστοποιήσουν) τη μοντελοποιημένη κατάσταση εμπειρίας (έξοδος ANN) για το συγκεκριμένο παίκτη. Τα εξατομικευμένα επίπεδα του Super Mario Bros που δημιουργήθηκαν δείχνουν ότι μπορεί να πραγματοποιηθεί ένας βασισμένος στην εμπειρία σκελετός δημιουργίας διαδικαστικού περιεχομένου, ο συναισθηματικός βρόχος μπορεί να κλείσει σε βιντεοπαιχνίδια και παρέχει μια καινοτόμο προσέγγιση στον έλεγχο και την προσαρμογή στα βιντεοπαιχνίδια. Η προτεινόμενη μεθοδολογία προσαρμογής ωστόσο, πρέπει να επικυρωθεί με τη βοήθεια πραγματικών παικτών σε αληθινές συνεδρίες gameplay όπου οι παίκτες καλούνται να παίξουν και να συγκρίνουν τυχαία δημιουργημένα επίπεδα με επίπεδα που είναι βελτιστοποιημένα για την μοντελοποιημένη εμπειρία ενός παίκτη. Αποτελέσματα από προηγούμενες μελέτες σε μικρό αριθμό παικτών αναδεικνύουν ότι το πλαίσιο προσαρμογής εξοντωτικής αναζήτησης είναι αποτελεσματικό στη δημιουργία επιπέδων που προτιμούνται από την πλειοψηφία των παικτών.

Η μέθοδος εξοντωτικής αναζήτησης που παρουσιάστηκε είναι κατάλληλη λόγω του σχετικά μικρού μεγέθους του χώρου αναζήτησης που εξερευνάται. Επικεντρωθήκαμε κυρίως στη συγχώνευση τεχνικών για τη δημιουργία αξιόπιστων μοντέλων εμπειρίας παίκτη. Μελλοντικές μελέτες θα μπορούσαν να ασχοληθούν με την κατασκευή και την επικύρωση γενικοτέρων μεθόδων για προσαρμογή παιχνιδιών, που είναι πιο αποτελεσματικές σε μεγαλύτερους χώρους αναζήτησης. Για παράδειγμα, μπορούν να χρησιμοποιηθούν εξελικτικές μέθοδοι για αυτόν το σκοπό. Προηγούμενες μελέτες απέδειξαν ότι υπάρχουν καλές προοπτικές για τις μετα-ευριστικές μεθόδους στην εξερεύνηση μεγάλων χώρων περιεχομένου ενσωματώνοντας το μηχανισμό προσαρμογής στη διαδικασία παραγωγής περιεχομένου.

1.2 Αντικείμενο Διπλωματικής

1.2.1 Σκοπός της Εργασίας

Σκοπός της εργασίας αυτής ήταν η ανάπτυξη εφαρμογής (σε C++) για την εξαγωγή των χαρακτηριστικών καρέ από τις ακολουθίες βίντεο που περιέχουν τις κινήσεις και οπτικές αντιδράσεις των παικτών που αναφέρθηκαν στην προηγούμενη παράγραφο. Η εφαρμογή που αναπτύχθηκε δέχεται ως είσοδο τα αρχεία βίντεο που κατέγραψαν την εμπειρία των παικτών αλλά και αρχεία κειμένου που παράγονται ως αποτελέσματα προηγουμένου προγράμματος, τα οποία περιέχουν σύνολα τιμών/συντεταγμένων χαρακτηριστικών του προσώπου (μάτια, στόμα κ.α.) για συγκεκριμένες χρονικές στιγμές(καρε;). Στη συνέχεια, οι τιμές αυτές αποθηκεύονται σε πίνακα δύο διαστάσεων για περεταίρω εξέταση. Ιδιαίτερα μας ενδιαφέρουν οι χρονικές στιγμές στις οποίες παρουσιάζεται απόκλιση της στάσης και κίνησης του κεφαλιού μεγαλύτερη του 10% σε σχέση με τη στιγμή κατά την οποία το κεφάλι του παίκτη βρίσκεται στην αρχική του θέση, σε αδρανή κατάσταση, και της οποίας οι τιμές των χαρακτηριστικών του κεφαλιού του παίκτη λαμβάνονται ως σημεία αναφοράς. Αποθηκεύοντας τις χρονικές αυτές στιγμές σε νέο πίνακα, μπορούμε με τη χρήση της βιβλιοθήκης OpenCV (λεπτομέρειες σε επόμενη παράγραφο) να επεξεργαστούμε τα αρχεία βίντεο και να εξάγουμε τα καρέ που αντιστοιχούν στις χρονικές στιγμές μεγάλης απόκλισης, σημειώνοντας σε αυτά και τις συντεταγμένες των χαρακτηριστικών του προσώπου.

Δόθηκε μεγαλύτερη σημασία στα καρέ και τα χαρακτηριστικά των χρονικών στιγμών αυτών καθώς σε συνδυασμό με τα χαρακτηριστικά gameplay, οι απότομες κινήσεις του κεφαλιού και οι έντονες οπτικές αντιδράσεις που συνδέονται με τις αποκλίσεις που εξετάζονται, αποδείχθηκαν οι βασικότερες στην εκτίμηση και την πρόβλεψη της συναισθηματικής κατάστασης των παικτών. Πρόκειται για τις χρονικές στιγμές που παρατηρούνται απότομες, αυθόρμητες ή απρόβλεπτες κινήσεις, όταν επομένως η ρευστότητα κίνησης κεφαλιού (FL) λαμβάνει υψηλότερες τιμές. Συνήθως παρουσιάζονται κατά τη διάρκεια μιας κρίσιμης κίνησης στο παιχνίδι, όπως σε ένα δύσκολο άλμα ή τη στιγμή της ήττας (όταν ο Μάριο χάνει μια ζωή).

Τα αποτελέσματα της εφαρμογής μπορούν να χρησιμοποιηθούν σε επόμενες μελέτες σε συνδυασμό με τεχνικές crowdsourcing για να πάρουμε χαρακτηρισμούς για το τι δείχνει καθένα από αυτά από ανώνυμους χρήστες και οι χαρακτηρισμοί αυτοί μπορούν να χρησιμοποιηθούν στα μοντέλα πρόβλεψης της συμπεριφοράς παίκτη, με τελικό σκοπό την παραγωγή δυναμικών, εξατομικευμένων επιπέδων που ο κάθε παίκτης θα βρει πιο ικανοποιητικά.

1.2.2 Οργάνωση Κειμένου

Στη συνέχεια της διπλωματικής εργασίας το κείμενο οργανώνεται ως εξής:

Περισσότερες λεπτομέρειες σχετικά με την προηγούμενη εφαργογή PoseGaze της οποίας τα αποτελέσματα δέχεται ως είσοδο η εφαρμογή παρουσιάζονται στην Παράγραφο 2. Στην Παράγραφο 3 αναφέρονται τα δεδομένα και τα εργαλεία που χρησιμοποιήθηκαν για την πραγματοποίηση της εξαγωγής των χαρακτηριστικών καρέ ιδαίτερου ενδιαφέροντος. Η εφαρμογή που παράγει τα επιθυμητά αποτελέσματα, ο σκοπός των επιμέρους τμημάτων της καθώς και ο γενικότερος αλγόριθμος στον οποίο βασίζεται αναλύονται στην Παράγραφο 4. Στην Παράγραφο 5 παρουσιάζονται τα αποτελέσματα της εφαρμογής και στην Παράγραφο 6 τα τελικά συμπεράσματα καθώς και περεταίρω μελλοντικές προοπτικές για το αντικείμενο της εργασίας.

2. Εφαρμογές PoseGaze και FacialFeat

Στην παράγραφο αυτή παρουσιάζονται οι εφαρμογές που αναπτύχθηκαν στο πλαίσιο προηγούμενης δουλειάς[1], τα αποτελέσματα των οποίων χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής dFrame, η οποία αποτελεί το κύριο μέρος της εργασίας αυτής. Οι εφαρμογές αυτές είναι οι:

-PoseGaze, η οποία χρησιμοποιείται κυρίως για την εξαγωγή χαρακτηριστικών σχετικών με τη στάση του κεφαλιού και το βλέμμα του παίκτη και

-FacialFeat, η οποία χρησιμοποιείται κυρίως για την εξαγωγή χαρακτηριστικών του προσώπου.

Η εκτέλεση των εφαρμογών αυτών γίνεται με είσοδο είτε κατευθείαν από τον εξοπλισμό καταγραφής (webcam) είτε με τη μορφή ήδη αποθηκευμένου αρχείου βίντεο (.avi). Τόσο η PoseGaze όσο και η FacialFeat χρησιμοποιούνται για την εξαγωγή των χαρακτηριστικών οπτικής αντίδρασης και κίνησης κεφαλιού του παίκτη. Τα δεδομένα που προκύπτουν καταγράφονται και αποθηκεύονται σε αρχεία κειμένου με τη μορφή μετρήσεων και κανονικοποιημένων συντεταγμένων για τα διάφορα σημεία του προσώπου τα οποία μας ενδιαφέρουν. Τα σημεία αυτά απεικονίζονται στην ακολουθία βίντεο αν αυτό επιλέξει ο χρήστης.

Τα αρχεία κειμένου που εξάγουν οι εφαρμογές αυτές αναλύονται σε επόμενη υποπαράγραφο και θα αποδειχθούν ιδιαίτερα χρήσιμα (ειδικά αυτά της FacialFeat) ως είσοδος στην εφαρμογή dFrame η οποία έχει ως αντικείμενο τον εντοπισμό και την εξαγωγή χαρακτηριστικών καρέ από τις ακολουθίες βίντεο που περιέχουν τη συμπεριφορά και τις αντιδράσεις των παικτών κατά τη διάρκεια των συνεδριών που αναφέρθηκαν στην πρώτη παράγραφο.

Στις υποπαραγράφους που ακολουθούν θα ασχοληθούμε με μια γενικότερη περιγραφή των εφαρμογών PoseGaze και FacialFeat και του τρόπου με τον οποίο αυτές λειτουργούν. Θα παρουσιαστούν επίσης παραδείγματα εξόδων (αρχείων κειμένου) που προκύπτουν από την εκτέλεση κάθε εφαρμογής καθώς και κάποια συμπεράσματα σχετικά με την πιθανότητα χρησιμοποίησης των αποτελεσμάτων αυτών στην πρόβλεψη των συναισθηματικών καταστάσεων της εμπειρίας παίκτη αλλά και στην υλοποίηση της εφαρμογής dFrame.

2.1 Περιγραφή Εφαρμογών

Η εφαρμογή που αναπτύχθηκε σε προηγούμενη μελέτη και αποτέλεσε βάση για την εργασία αυτή ονομάζεται “FacialFeat” και έχει ως αντικείμενο την καταγραφή των χαρακτηριστικών κίνησης κεφαλιού και οπτικής αντίδρασης παίκτη, τα οποία και εξάγει σε μορφή αρχείου κειμένου για ευκολότερη επεξεργασία τους. Χρησιμοποιήθηκε στα πειράματα της [1] με σκοπό τη μοντελοποίηση της εμπειρίας παίκτη για την πρόβλεψη της συμπεριφοράς των τριων συναισθηματικών καταστάσεων, προσήλωσης, εκνευρισμού και πρόκλησης.

Η FacialFeat σε συνδυασμό με την PoseGaze, η οποία χρησιμοποιείται για τον υπολογισμό των διανυσμάτων Στάσης(Pose) και Βλέμματος(Gaze) λειτουργεί χρησιμοποιώντας τον διαθέσιμο εξοπλισμό καταγραφής (συνήθως ενσωματωμένες κάμερες σε φορητούς υπολογιστές, webcams κ.α.) για τη συλλογή των χαρακτηριστικών του προσώπου και κίνησης που μας ενδιαφέρουν αλλά και του διανύσματος Στάσης-Βλέμματος, το οποίο βοηθάει στην καταγραφή της εκτροπής και της κλίσης του κεφαλιού καθώς και των πιο απότομων κινήσεων και οπτικών αντιδράσεων, οι οποίες όπως αναφέραμε παραπάνω παρουσιάζουν ιδιαίτερο ενδιαφέρον στην μοντελοποίηση της εμπειρίας παίκτη και αποτελούν το κύριο αντικείμενο της εργασίας αυτής.

Η PoseGaze μπορεί να εφαρμοστεί σε συνδυασμό με είσοδους του χρήστη με τη μορφή των εξής προαιρετικών, μη τοποθετημένων παραμέτρων εισόδου (arguments):

(a) -*inputVideo* arg: δίνει τη δυνατότητα στην εφαρμογή να επεξεργαστεί κάποιο άρχειο βίντεο, αντί της κάμερας. Σαν παράμετρος στην εφαρμογή δίνεται το όνομα του αρχείου βίντεο προς επεξεργασία. Η επιλογή αυτή διευκολύνει την εξέταση παλαιότερου υλικού με αντιδράσεις παικτών, επεκτείνοντας τη χρηστικότητα της PoseGaze πέρα από την συλλογή των δεδομένων που λαμβάνονται από τον εξοπλισμό καταγραφής όσο ακόμα τρέχει η εφαρμογή.

(b) -*referenceFrame* arg: δίνει τη δυνατότητα στην εφαρμογή να χρησιμοποιήσει συγκεκριμένο καρέ που βρίσκεται στην κατοχή του χρήστη σαν σημείο αναφοράς για τα χαρακτηριστικά του προσώπου του παίκτη, τις συντεταγμένες τους και το διάνυσμα Στάσης-Βλέμματος (PoseGaze). Σαν παράμετρος στην εφαρμογή δίνεται το όνομα του αρχείου εικόνας του συγκεκριμένου καρέ. Η χρήση ενός καρέ αναφοράς θα αποδειχθεί πολύ σημαντική για την επόμενη εφαρμογή και τον εντοπισμό των καρέ με τις μεγαλύτερες αποκλίσεις καθώς παρέχει ένα μέτρο σύγκρισης και διευκολύνει αρκετά την αναζήτηση των καρέ αυτών.

(c) -*standaloneImage* arg: δίνει τη δυνατότητα στην εφαρμογή να επεξεργαστεί συγκεκριμένο καρέ αντί για ολόκληρο αρχείο βίντεο. Σαν παράμετρος στην εφαρμογή δίνεται το όνομα του αρχείου εικόνας του συγκεκριμένου καρέ. Η λειτουργία θα μπορούσε να είναι ιδιαίτερα χρήσιμη στην εξέταση των διαθέσιμων καρέ που αντιστοιχούν σε χρονικές στιγμές κρίσιμων συμβάντων στο παιχνίδι (όπως κάποιο δύσκολο άλμα ή η ήττα) και επομένως στη συγχώνευση χαρακτηριστικών gameplay και συμπεριφοράς, η οποία και αποδείχθηκε πολύ ικανό μέσο πρόβλεψης των συναισθηματικών καταστάσεων.

(d) -*attentiveness*: δίνει τη δυνατότητα στην εφαρμογή να παράγει και να απεικονίζει τιμές προσοχής (attentiveness).

(e) -*attentivenessTransmitter:* δίνει τη δυνατότητα στην εφαρμογή να παράγει, να απεικονίζει και να αποστέλλει τιμές προσοχής. Η θύρα/IP λαμβάνεται από το αρχείο κειμένου POSE_GAZE.

Το αρχείο POSE_GAZE περιέχει τις απαραίτητες προδιαγραφές για την εκτέλεση της εφαρμογής FacialFeat και μπορεί να έχει τη μορφή:

```
scale:
1
video input:
1
Dest IP:
localhost
Port:
1111
Video Display:
1
Video Output:
0
FPS:
10
File Output:
1
```

όπου:

1. Scale είναι η κλίμακα των διαστάσεων του βίντεο και FPS (Frames Per Second, καρέ ανά δευτερόλεπτο) είναι ο ρυθμός των καρέ της αποθηκευμένης ακολουθίας βίντεο.

2 .Η μεταβλητή Video input (βίντεο εισόδου), αφορά τον τρόπο λειτουργίας της εφαρμογής. Πιο συγκεκριμένα, για εναλλαγή μεταξύ των λειτουργιών επεξεργασίας αρχείων βίντεο τύπου avi και εισόδου από webcam, αντιστοιχείται η τιμή 0 στην είσοδο αρχείου avi και η τιμή 1 στην είσοδο live webcam.

Λειτουργία AVI

Βήμα 1: Ο χρήστης πρέπει να ανοίξει το αρχείο κειμένου και να βεβαιωθεί ότι η μεταβλητή Video input έχει την τιμή 0/false, αλλιώς πρέπει να αλλάξει την τιμή, να αποθηκεύσει και να κλείσει το αρχείο.
Βήμα 2: Ο χρήστης εκτελεί το αρχείο .exe. Σε αυτό το σημείο θα ζητηθεί από το χρήστη το όνομα του αρχείου avi που θέλει να τρέξει.

Λειτουργία WebCam

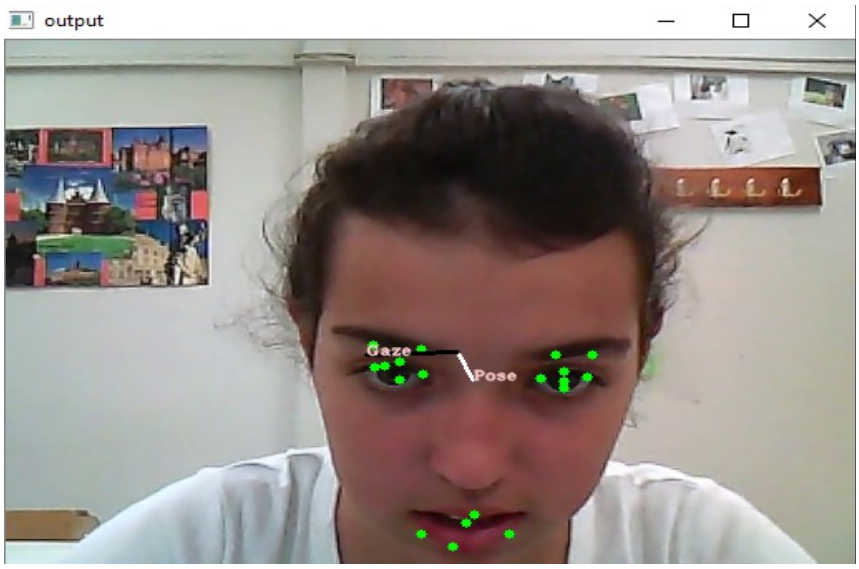
Βήμα 1: Ο χρήστης πρέπει να ανοίξει το αρχείο κειμένου και να βεβαιωθεί ότι η μεταβλητή Video input έχει την τιμή 1/true, αλλιώς πρέπει να αλλάξει την τιμή, να αποθηκεύσει και να κλείσει το αρχείο.
Βήμα 2: Εκτέλεση του αρχείου .exe. Το παράθυρο εξόδου εμφανίζεται σαν ορθογώνια έξοδος βίντεο, δείχνοντας την είσοδο από τη webcam.
Βήμα 3: Ο χρήστης τοποθετεί το πρόσωπο του μπροστά στην οθόνη όπως στα ενδεικτικά βίντεο που συνοδεύουν την εφαρμογή
Βήμα 4: Έχοντας επιλέξει με το ποντίκι το παράθυρο εξόδου και τοποθετήσει το πρόσωπο σωστά σε αυτό, ο χρήστης πατά ENTER κοιτώντας στο κέντρο, μένει σχετικά ακίνητος και παρακολουθεί την έξοδο.

3. Η μεταβλητή Dest IP προσδιορίζει την διεύθυνση IP του προορισμού για την αποστολή των δεδομένων OSC (παράμετροι στάσης κεφαλιού και βλέμματος, συντεταγμένες χαρακτηριστικών προσώπου και τιμές προσοχής).

4. Η μεταβλητή Dest port προσδιορίζει τη θύρα της μηχανής προορισμού.

5. Η μεταβλητή Video display (απεικόνιση βίντεο) δίνει τη δυνατότητα στο χρήστη να έχει οπτική ανατροφοδότηση της διαδικασίας, όταν παίρνει την τιμή 1/true. Αν αυτό δεν ενδιαφέρει το χρήστη αφήνει την τιμή της μεταβλητής στο 0/false.

6. Η μεταβλητή Video output (βίντεο εξόδου), αποθηκεύει ένα αρχείο .avi της διαδικασίας όπου φαίνονται οι συντεταγμένες των χαρακτηριστικών και η κατεύθυνση στάσης κεφαλιού/βλέμματος. Η λευκή γραμμή απεικονίζει τη στάση του κεφαλιού και η μαύρη το βλέμμα, όπως φαίνεται στην εικ. 8.



Εικ 8: Στιγμιότυπο αρχείου βίντεο εξόδου, όπου με πράσινο απεικονίζονται τα χαρακτηριστικά του προσώπου βάσει των συντεταγμένων τους, καθώς και τα διανύσματα στάσης κεφαλιού (άσπρο) και βλέμματος (μαύρο).

7. Η μεταβλητή File output (αρχείο εξόδου), αποθηκεύει ένα αρχείο κειμένου .txt. Το όνομα του αρχείου έχει τη μορφή output_DATETIME.text (έξοδος-ημερομηνία και ώρα). Στην πρώτη στήλη του αρχείου δίνεται ένας ακέραιος αριθμός που αντιπροσωπεύει τα milliseconds αφού ξεκινήσει η διαδικασία και στις επόμενες στήλες ακολουθούν τα δεδομένα. Τα αποτελέσματα του αρχείου αυτού εξετάζονται περεταίρω στην επόμενη υποπαράγραφο.

Για την υλοποίηση της εφαρμογής PoseGaze χρησιμοποιήθηκε η βιβλιοθήκη OpenCV (Open source Computer Vision) η οποία θα παρουσιαστεί με περισσότερες λεπτομέρειες στην επόμενη παράγραφο. Η FacialFeat καταγράφει τις τιμές για το βλέμμα του παίκτη ως τις συντεταγμένες τεσσάρων σημείων των ματιών (πάνω, κάτω, δεξιά, αριστερά). Με παρόμοιο τρόπο καταγράφονται από την PoseGaze τα υπόλοιπα χαρακτηριστικά του προσώπου και το διάνυσμα Στάσης-Βλέμματος. Η εφαρμογή τελικά μας δίνει σαν έξοδο τις θέσεις 19 χαρακτηριστικών του προσώπου για κάθε καρέ, καθώς και μια ετικέτα που δηλώνει την κατάσταση προσοχής του παίκτη προς το υλικό στο οποίο εκτίθεται.

2.2 Αποτελέσματα της Εφαρμογής

2.2.1 Αποτελέσματα Εφαρμογής PoseGaze

Η εκτέλεση της εφαρμογής PoseGaze γίνεται με βασικότερο σκοπό την καταγραφή δεδομένων σχετικών με τα διανύσματα Στάσης-Βλέμματος και την απεικόνιση των διανυσμάτων αυτών σε συνδυασμό με τα χαρακτηριστικά προσώπου που καταγράφονται από την FacialFeat. Το αρχείο κειμένου που παράγεται ως έξοδος της εφαρμογής περιέχει ένα σύνολο από καταχωρήσεις που παρουσιάζουν τις τιμές που μας ενδιαφέρουν για κάθε χρονική στιγμή. Μια τέτοια καταχώρηση έχει

την εξής μορφή(οι τιμές των μετρήσεων είναι τυχαία επιλεγμένες):

```
0hrs::0mins::0secs::946millisecs::928microsecs::399nanosecs
(poseMetric: Right-Up
 gazeMetric: Left-Up
 poseLengthMetric: 0.0950739
 gazeLengthMetric: 18.8873
 distanceFromMonitorMetric: 0.109756
 lEyeVerticalMetric: 2.56827
 rEyeVerticalMetric: 1.687
 mouthVerticalMetric: 5.39033
 mouthHorizontalMetric: 1.24602
 lEyeInnerToLEyebrowInnerMetric: 7.44965
 lEyeInnerToLEyebrowOuterMetric: 3.54611
 rEyeInnerToREyebrowInnerMetric: 5.54068
 rEyeInnerToREyebrowOuterMetric: 5.96489)
```

Είναι προφανές ότι στην πρώτη γραμμή κάθε καταχώρησης δίνεται η χρονική στιγμή του καρέ, διαιρεμένη σε ώρες, λεπτά, δευτερόλεπτα, milliseconds, microseconds και nanoseconds. Στη συνέχεια παρουσιάζονται οι τιμές των χαρακτηριστικών που απαιτούνται για την απεικόνιση των διανυσμάτων:

-poseMetric: ο προσανατολισμός του διανύσματος της στάσης του κεφαλιού (στο παράδειγμα αυτό Δεξιά-Πάνω).

-gazeMetric: ο προσανατολισμός του διανύσματος του βλέμματος (στο παράδειγμα αυτό Αριστερά-Πάνω).

-poseLengthMetric: η μέτρηση για το μήκος του διανύσματος της στάσης του κεφαλιού.

-gazeLengthMetric: η μέτρηση για το μήκος του διανύσματος του βλέμματος.

-distanceFromMonitorMetric: η μέτρηση για την απόσταση του παίκτη από την οθόνη.

-lEyeVerticalMetric: η μέτρηση για την κάθετη απόσταση του αριστερού ματιού.

-rEyeVerticalMetric: η μέτρηση για την κάθετη απόσταση του δεξιού ματιού.

-mouthVerticalMetric: η μέτρηση για τα σημεία του στόματος, κάθετα.

-mouthHorizontalMetric: η μέτρηση για τα σημεία του στόματος, οριζόντια.

-lEyeInnerToLEyebrowInnerMetric: η μέτρηση για την εσωτερική απόσταση του αριστερού ματιού από το αριστερό φρύδι.

-lEyeInnerToLEyebrowOuterMetric: η μέτρηση για την εξωτερική απόσταση του αριστερού ματιού από το αριστερό φρύδι.

-rEyeInnerToREyebrowInnerMetric: η μέτρηση για την εσωτερική απόσταση του δεξιού ματιού από το δεξί φρύδι.

-rEyeInnerToREyebrowOuterMetric: η μέτρηση για την εξωτερική απόσταση του δεξιού ματιού από το δεξί φρύδι.

Ο συνδυασμός των δεδομένων αυτών μας δίνει μια πιο ακριβή εικόνα για τα διανύσματα Στάσης-Βλέμματος όπως αυτά απεικονίζονται στο παράδειγμα της εικόνας 8.

2.2.2 Αποτελέσματα Εφαρμογής FacialFeat

Η εκτέλεση της εφαρμογής FacialFeat γίνεται με βασικότερο σκοπό την καταγραφή δεδομένων σχετικών με τα χαρακτηριστικά του προσώπου και τις συντεταγμένες τους και την απεικόνισή τους στη συνέχεια σε συνδυασμό με τα διανύσματα Στάσης κεφαλιού και Βλέμματος μέσω της PoseGaze.

Πιο συγκεκριμένα, το αρχείο κειμένου περιέχει μετρήσεις σχετικές με:

-το διάνυσμα Βλέμματος (eye gaze): κάθετες και οριζόντιες κινήσεις (2 αριθμοί τύπου float),

-το διάνυσμα Στάσης κεφαλιού (Head Pose): κλίση, εκτροπή (2 αριθμοί τύπου float). Η κλίση και η εκτροπή παρουσιάζονται σαν κανονικοποιημένοι floats.

-την απόσταση από την οθόνη: αριθμός τύπου float που δείχνει την απόσταση του χρήστη από την οθόνη. Τιμές που τείνουν στη μονάδα δηλώνουν σταθερή απόσταση, τιμές μικρότερες της μονάδας δηλώνουν απομάκρυνση από την οθόνη και τιμές μικρότερες της μονάδας δηλώνουν προσέγγιση της οθόνης.

-την περιστροφή κεφαλιού: η περιστροφή του κεφαλιού μετράται σε μοίρες (1 αριθμός τύπου float).

-τα χαρακτηριστικά του προσώπου σε κανονικοποιημένες συντεταγμένες, οι οποίες λαμβάνονται με την ακόλουθη σειρά:

γραμμή (δεξί μάτι, δεξιά γωνία)
στήλη (δεξί μάτι, δεξιά γωνία)
γραμμή (δεξί μάτι, αριστερή γωνία)
στήλη (δεξί μάτι, αριστερή γωνία)

γραμμή (αριστερό μάτι, δεξιά γωνία)
στήλη (αριστερό μάτι, δεξιά γωνία)
γραμμή (αριστερό μάτι, αριστερή γωνία)
στήλη (αριστερό μάτι, αριστερή γωνία)

γραμμή (δεξί μάτι, κέντρο)
στήλη (δεξί μάτι, κέντρο)
γραμμή (αριστερό μάτι, κέντρο)
στήλη (αριστερό μάτι, κέντρο)

γραμμή (δεξί μάτι, μεσαίο σημείο στο άνω βλέφαρο)
στήλη (δεξί μάτι, μεσαίο σημείο στο άνω βλέφαρο)
γραμμή (δεξί μάτι, μεσαίο σημείο στο κάτω βλέφαρο)
στήλη (δεξί μάτι, μεσαίο σημείο στο κάτω βλέφαρο)

γραμμή (αριστερό μάτι, μεσαίο σημείο στο άνω βλέφαρο)
στήλη (αριστερό μάτι, μεσαίο σημείο στο άνω βλέφαρο)
γραμμή (αριστερό μάτι, μεσαίο σημείο στο κάτω βλέφαρο)
στήλη (αριστερό μάτι, μεσαίο σημείο στο κάτω βλέφαρο)

γραμμή (δεξιά γωνία του στόματος)
στήλη (δεξιά γωνία του στόματος)
γραμμή (αριστερή γωνία του στόματος)

στήλη (αριστερή γωνία του στόματος)

γραμμή (ανώτερο σημείο στόματος)
στήλη (ανώτερο σημείο στόματος)
γραμμή (κατώτερο σημείο στόματος)
στήλη (κατώτερο σημείο στόματος)

γραμμή (άκρη της μύτης)
στήλη (άκρη της μύτης)

γραμμή (εξωτερική γωνία δεξιού φρυδιού)
στήλη (εξωτερική γωνία δεξιού φρυδιού)
γραμμή (εσωτερικη γωνία δεξιού φρυδιού)
στήλη (εσωτερική γωνία δεξιού φρυδιού)

γραμμή (εσωτερική γωνία αριστερού φρυδιού)
στήλη (εσωτερική γωνία αριστερού φρυδιού)
γραμμή (εξωτερική γωνία αριστερού φρυδιού)
στήλη (εξωτερική γωνία αριστερού φρυδιού)

Καταστάσεις προσοχής:
αποστέλλονται ως γεγονότα, τη στιγμή που συμβαίνουν.
Ετικέτες: ATTENTIVE=4 (προσοχή)
DISTRACTED=2 (Κατά πόσο αποσπάστηκε ο χρήστης)
STRUGGLING TO READ=1 (Διαβάζει με μεγάλη δυσκολία)
TIRED=5 (Κουρασμένος)

Παρουσιάζεται ως παράδειγμα ένα τμήμα αρχείου κειμένου που εξήγαγε η εφαρμογή:

344;0.000000;0.000000;0.000000;0.000000;1.000000;-1.763285;.....;213.000000;195.000000
453;6.551723;0.000000;0.009023;0.000000;1.000000;-2.424695;.....;210.478531;194.198578
547;2.488049;0.000000;-0.005787;0.000000;1.000000;-3.294976;.....;210.054092;193.415298
610;4.529636;0.000000;-0.000154;0.001632;1.015385;-3.222243;.....;209.401031;192.982742
703;4.481786;0.000000;-0.002222;0.003427;1.015385;-3.254266;.....;209.684830;192.871719
.
.
.
2532;8.277664;0.000000;-0.074361;0.102365;0.933333;-0.533510;.....;156.993256;191.842087
2594;6.197838;0.000000;-0.083057;0.102287;0.950000;-1.524707;.....;156.913132;191.315201
2688;8.184307;0.000000;-0.066228;0.107590;0.966667;-2.497873;.....;156.442047;191.406998

Στην πρώτη στήλη (τιμές 453, 547 κ.ο.κ.) εμφανίζονται οι τιμές των milliseconds για τα καρέ που αντιπροσωπεύουν. Στο συγκεκριμένο παράδειγμα, η ακολουθία βίντεο είχε διάρκεια 2.7 δευτερόλεπτα, όπως φαίνεται και στο αρχείο κειμένου, με το πρώτο καρέ να αντιστοιχεί στα 453 milliseconds και το τελευταίο στα 2688 milliseconds.

Το υπόλοιπο τμήμα κάθε γραμμής παρουσιάζει τις κανονικοποιημένες συντεταγμένες των χαρακτηριστικών του προσώπου του παίκτη με τη μορφή που αναφέρθηκε.

2.2.3 Συμπεράσματα

Στη συγκεκριμένη εργασία, μας ενδιαφέρουν περισσότερο τα αρχεία κειμένου που εξάγει η εφαρμογή FacialFeat. Αυτά θα αποτελέσουν, μαζί με τα βίντεο στα οποία αντιστοιχούν, τις εισόδους της νέας εφαρμογής dFrame, σκοπός της οποίας είναι ο εντοπισμός και η επεξεργασία των χαρακτηριστικών καρέ που παρουσιάζουν τις μεγαλύτερες αποκλίσεις από τα σημεία αναφοράς στις συντεταγμένες των διαφόρων χαρακτηριστικών του προσώπου του παίκτη. Τα αποτελέσματα της PoseGaze, ενώ διευκολύνουν αρκετά την ανάλυση των οπτικών αντιδράσεων των παικτών, δεν είναι απαραίτητα για την νέα εφαρμογή dFrame. Η εύρεση των καρέ που μας ενδιαφέρουν γίνεται βάσει συγκρίσεων με σημεία αναφοράς και αφορούν συγκεκριμένα στιγμιότυπα των βίντεο των συνεδριών, ενώ τα δεδομένα που εξάγει η εφαρμογή PoseGaze αφορούν τα διανύσματα Στάσης κεφαλιού και Βλέμματος. Τα στοιχεία αυτά θα χρησίμευαν περισσότερο σε κάποια μελέτη της γενικότερης συμπεριφοράς του παίκτη καθ' όλη τη διάρκεια της συνεδρίας παιχνιδιού αλλά και τον ακριβέστερο υπολογισμό της ρευστότητας κίνησης (FL), γεγονός πολύ σημαντικό για την ακριβή πρόβλεψη της συναισθηματικής κατάστασης, αλλά όχι απαραίτητο στην εξαγωγή των χαρακτηριστικώνκαρέ των πιο απότομων κινήσεων του παίκτη.

Τα αποτελέσματα τόσο της PoseGaze όσο και της FacialFeat είναι απαραίτητα για την κωδικοποίηση και ευκολότερη επεξεργασία των χαρακτηριστικών του προσώπου του παίκτη όπως αυτά απεικονίζονται στην εικόνα 8. Βάσει αυτών διευκολύνεται η ανάλυση της κίνησης του κεφαλιού και των οπτικών αντιδράσεων του παίκτη και επομένως η μοντελοποίηση της συμπεριφοράς του με σκοπό την πρόβλεψη των τριών συναισθηματικών καταστάσεων (προσήλωση, εκνευρισμός, πρόκληση). Ιδιαίτερο ενδιαφέρον θα παρουσίαζε και η εξέταση και σύγκριση των μετρήσεων που θα προέκυπταν από την εκτέλεση των εφαρμογών αυτών για πολλούς διαφορετικούς παίκτες και συνεδρίες παιχνιδιού, συνδυασμένη με αντίστοιχες μετρήσεις και δεδομένα χαρακτηριστικών gameplay και περιεχομένου του παιχνιδιού. Σκοπός μιας τέτοιας πιθανής μελλοντικής μελέτης θα ήταν μια περεταίρω στατιστική ανάλυση των διαφορών της συμπεριφοράς και των αντιδράσεων των παικτών στα εξωτερικά ερεθίσματα και τα πιο κρίσιμα συμβάντα του παιχνιδιού με το οποίο ασχολούμαστε (για το Super Mario Bros συγκεκριμένα, συμβάντα όπως το ποδοπάτημα ενός εχθρού από το Μάριο, το θάνατο του Μάριο, ένα δυσκολο άλμα κ.α.).

Θα ήταν δυνατή πιθανότατα και η γενίκευση των πειραμάτων και των αποτελεσμάτων τους σε άλλα είδη παιχνιδιών παρόμοιας φιλοσοφίας, gameplay και συμπεριφοράς που επιδεικνύει ο παίκτης, όπως τα shooters πρώτου προσώπου (FPS). Τα FPS μπορούν να προκαλέσουν ανάλογες αντιδράσεις στον παίκτη σε αντίστοιχα κρίσιμα συμβάντα, όπως παραδείγματος χάριν όταν ο παίκτης σκοτώσει κάποιον αντίπαλο, πετυχαίνοντας τον στο κεφάλι (headshot), όταν ο χαρακτήρας του παίκτη τραυματίζεται από κάποιον αντίπαλο ή πεθαίνει, όταν εκτελεί κάποια δύσκολη βολή κ.ο.κ. Είναι ευκολότερη η εφαρμογή της διαδικασίας και των πειραμάτων με τα οποία ασχολούμαστε σε αυτού του τύπου παιχνίδια, όπως παιχνίδια πλατφόρμας, δράσης, shooters και άλλα, τα οποία βασίζουν το gameplay και το περιεχόμενό τους κυρίως στα αντανακλαστικά του παίκτη και τις αντιδράσεις του σε κρίσιμα συμβάντα που διαδέχονται συνεχώς το ένα το άλλο, με σκοπό να κρατήσουν την προσοχή του παίκτη αμείωτη. Επειδή αυτά τα παιχνίδια βασίζονται κυρίως στα αντανακλαστικά του παίκτη, είναι αναμενόμενο ότι ο αριθμός των διαφορετικών χαρακτηριστικών gameplay και περιεχομένου που χρησιμοποιούμε θα είναι , αν όχι μικρός, εύκολα διαχειρίσιμος αλλά και ότι οι αντιδράσεις του παίκτη, οι απαλές και απότομες κινήσεις του κεφαλιού και οι μεταβολές στα χαρακτηριστικά του προσώπου του θα είναι περισσότερες και θα εντοπίζονται εύκολα. Αντίθετα, η εφαρμογή της διαδικασίας και των πειραμάτων που πραγματοποιήθηκαν, και επομένως η μοντελοποίηση της συμπεριφοράς του παίκτη και η ακριβής πρόβλεψη της συναισθηματικής του κατάστασης θα ήταν αισθητά δυσκολότερη σε παιχνίδια με πιο πολύπλοκο ή αργό gameplay. Βιντεοπαιχνίδια που ανήκουν σε είδη όπως στρατηγικής (RTS), περιπέτειας (adventure) ή ρόλων (RPG) θα απαιτούσαν την ενασχόληση και επιλογή ενός πολύ

μεγαλύτερου αριθμού χαρακτηριστικών gameplay και περιεχομένου, καθώς κάθε κρίσιμο συμβάν μπορεί να αποτελείται από πολλά μικρο-συμβάντα και να απευθύνεται περισσότερο στην κρίση του παίκτη, στη σκέψη και στη σωστή χρήση των επιλογών που έχει στη διάθεση του από ότι στα αντανακλαστικά του, με αποτέλεσμα και οι αντιδράσεις των παικτών να είναι πιο ομαλές και δύσκολα μετρήσιμες. Η επιλογή όμως των παιχνιδιών του πρώτου τύπου, με γρήγορο gameplay και ανταμοιβή των γρήγορων αντανακλαστικών του παίκτη, είναι πολύ πιο αποδοτική καθώς τα αποτελέσματα που θα εξαχθούν έχουν πολύ μεγαλύτερη εφαρμογή στη σύγχρονη αγορά. Τα shooters και τα γρήγορα παιχνίδια πλατφόρμας είναι και τα περισσότερο δημοφιλή για τον μέσο παίκτη, τόσο σε υπολογιστές όσο και σε κινητά τηλέφωνα, ενώ το κοινό των πιο αργών και πολύπλοκων βιντεοπαιχνιδιών είναι σημαντικά μικρότερο.

3. Εργαλεία και Δεδομένα για την Υλοποίηση της Εφαρμογής dFrame

Στην Παράγραφο αυτή παρουσιάζονται κάποιες γενικές πληροφορίες για τα εργαλεία πραγραμματισμού που χρησιμοποιήθηκαν στην ανάπτυξη του πηγαίου κώδικα σχετικού με την επεξεργασία εισόδων αρχείων κειμένου αλλά και ακολουθιών βίντεο και εικόνων μέσω τεχνικών οράσεως υπολογιστών (Computer Vision) και τελικά την ολοκληρωμένη υλοποίηση της εφαρμογής dFrame. Σκοπός της εφαρμογής dFrame είναι η εξαγωγή χαρακτηριστικών καρέ από ακολουθίες βίντεο με παίκτες Super Mario Bros, καρέ που απεικονίζουν τις πιο απότομες ή έντονες κινήσεις και αντιδράσεις του παίκτη, υπολογίζοντας τις βάσει του ποσοστού απόκλισης από συντεταγμένες που λαμβάνονται ως σημεία αναφοράς για κάθε επιμέρους χαρακτηριστικό του προσώπου του παίκτη (διάφορα σημεία για τα μάτια, τη μύτη, το στόμα). Για την τελική πραγματοποίηση της εφαρμογής αυτής χρησιμοποιήθηκαν εργαλεία που περιλαμβάνουν:

-το ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse,

-η γλώσσα προγραμματισμού C/C++ και

-οι επιπλέον βιβλιοθήκες που χρειάστηκαν (OpenCV),

τα οποία θα παρουσιαστούν πιο αναλυτικά στην επόμενη υποπαράγραφο, όπου θα αναφέρουμε και τους λόγους για τους οποίους αυτά επιλέχθηκαν.

Εξετάζεται επίσης περεταίρω και ο τρόπος χρήσης των αρχείων κειμένου που περιέχουν τις κανονικοποιημένες συντεταγμένες των χαρακτηριστικών προσώπου του παίκτη, όπως τα εξήγαγε για κάθε διαφορετικό παίκτη/ακολουθία βίντεο η εφαρμογή FacialFeat που αναλύθηκε παραπάνω, στην υλοποίηση και εκτέλεση της εφαρμογής dFrame. Εκτός από αυτά τα αρχεία κειμένου, η εφαρμογή dFrame δέχεται ως είσοδο και τα αρχεία βίντεο που αντιστοιχούν στη συνεδρία παιχνιδιού για την οποία λήφθηκαν οι μετρήσεις. Επίσης, όπως θα δούμε και στη συνέχεια, η dFrame προαιρετικά μπορεί να δεχθεί σαν παράμετρο ένα όνομα για νέο αρχείο κειμένου, το οποίο θα περιλαμβάνει τα αριθμητικά δεδομένα του αρχείου κειμένου εισόδου (χρονικές στιγμές και τις σχετικές τιμές των συντεταγμένων των χαρακτηριστικών του προσώπου) που αντιστοιχούν στα χαρακτηριστικά καρέ που μας απασχολούν.

3.1 Εργαλεία που Χρησιμοποιήθηκαν

Στις επόμενες υποενότητες παρουσιάζονται κάποιες γενικές πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής dFrame, πιο συγκεκριμένα για το Eclipse IDE, τη γλώσσα C++ και τη βιβλιοθήκη OpenCV (Open source Computer Vision, ανοιχτός κώδικας όρασης υπολογιστών).

3.1.1 Γλώσσες Προγραμματισμού C και C++

Η εφαρμογή dFrame αναπτύχθηκε στη γλώσσα προγραμματισμού C++ (με κάποια στοιχεία από C). Σε αυτήν την υποπαράγραφο παρουσιάζονται κάποιες γενικές πληροφορίες για τις γλώσσες αυτές.

Η Γλώσσα Προγραμματισμού C

Η C είναι μια διαδικαστική γλώσσα προγραμματισμού γενικής χρήσης, η οποία αναπτύχθηκε αρχικά, μεταξύ του 1969 και του 1973, από τον Ντένις Ρίτσι στα εργαστήρια AT&T Bell Labs για να χρησιμοποιηθεί για την ανάπτυξη του λειτουργικού συστήματος UNIX. Όπως οι περισσότερες διαδικαστικές γλώσσες προγραμματισμού που ακολουθούν την παράδοση της ALGOL, η C έχει δυνατότητες δομημένου προγραμματισμού και επιτρέπει τη χρήση αναδρομής (αλλά όχι και εμφωλευμένων συναρτήσεων), ενώ ο στατικός ορισμός του τύπου των μεταβλητών που επιβάλλει, προλαμβάνει πολλά σφάλματα κατά την χρήση τους. Ο σχεδιασμός της περιλαμβάνει δομές που μεταφράζονται αποδοτικά σε τυπικές εντολές μηχανής (machine instructions) και για αυτό το λόγο χρησιμοποιείται συχνά σε εφαρμογές που παλιότερα γράφονταν σε συμβολική γλώσσα (assembly language). Αυτό ακριβώς το χαρακτηριστικό της, που έχει σαν συνέπεια και την αυξημένη ταχύτητα εκτέλεσης των εφαρμογών που γράφονται σε αυτή, καθώς και το γεγονός ότι είναι διαθέσιμη στα περισσότερα σημερινά λειτουργικά συστήματα, συνέβαλε κατά πολύ στην καθιέρωση της και την χρήση της για ανάπτυξη λειτουργικών συστημάτων και λοιπών προγραμμάτων συστήματος (system software), αλλά και απλών εφαρμογών.

Η C συγκαταλέγεται πλέον στις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού όλων των εποχών και πολλές νεότερες γλώσσες έχουν επηρεαστεί άμεσα ή έμμεσα από αυτήν, συμπεριλαμβανομένων των C++ (την οποία θα δούμε στη συνέχεια), C#, D, Go, Java, JavaScript, Limbo, LPC, Perl, PHP, Python, καθώς και του κελύφους C (C shell) του Unix. Κάποιες από αυτές τις γλώσσες έχουν επηρεαστεί κυρίως στη σύνταξη τους, με το σύστημα τύπων, τα μοντέλα δεδομένων και το νόημα των εκφράσεων τους να διαφέρουν σημαντικά από την C. Η C++, ειδικά, ξεκίνησε σαν προεπεξεργαστής της C, αλλά έχει εξελιχθεί πλέον σε μια αντικειμενοστραφή γλώσσα, που αποτελεί υπερσύνολο της C.

Η C είναι μια σχετικά μινιμαλιστική γλώσσα προγραμματισμού. Ανάμεσα στους σχεδιαστικούς στόχους που έπρεπε να καλύψει η γλώσσα περιλαμβανόταν το ότι θα μπορούσε να μεταγλωττιστεί άμεσα με τη χρήση μεταγλωττιστή ενός περάσματος (single-pass compiler) — με άλλα λόγια, ότι θα απαιτούνταν μόνο ένας μικρός αριθμός από εντολές σε γλώσσα μηχανής για κάθε βασικό στοιχείο της, χωρίς εκτεταμένη υποστήριξη στον χρόνο εκτέλεσης. Ως αποτέλεσμα, είναι δυνατό να γραφτεί κώδικας σε C σε χαμηλό επίπεδο προγραμματισμού με ακρίβεια ανάλογη της συμβολικής γλώσσας, στην πραγματικότητα η C ορισμένες φορές αποκαλείται (και χωρίς να υπάρχει πάντα αντιπαράθεση) «συμβολική γλώσσα υψηλού επιπέδου» («high-level assembly») ή «φορητή συμβολική γλώσσα» («portable assembly»). Επίσης, γίνονται αναφορές στη C ως γλώσσα προγραμματισμού μεσαίου επιπέδου[15].

Η Γλώσσα Προγραμματισμού C++

Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού Ηλεκτρονικών Υπολογιστών. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών από γλώσσες υψηλού και χαμηλού επιπέδου. Είναι μια μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων, με τύπους. Υποστηρίζει δομημένο, αντικειμενοστραφή και γενικό προγραμματισμό.

Η γλώσσα αναπτύχθηκε από τον Μπιάρνε Στρούστρουπ το 1979 στα εργαστήρια Bell της AT&T,

ως βελτίωση της ήδη υπάρχουσας γλώσσας προγραμματισμού C, και αρχικά ονομάστηκε "C with Classes", δηλαδή C με Κλάσεις. Μετονομάστηκε σε C++ το 1983. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων, και ακολούθησαν, μεταξύ άλλων, εικονικές συναρτήσεις, υπερφόρτωση τελεστών, πολλαπλή κληρονομικότητα, πρότυπα κ.α.[16]

Πιο συγκεκριμένα, η C++:

- είναι μια ανοιχτή, ISO-τυποποιημένη γλώσσα από το 1998,
- είναι μια μεταγλωττιζόμενη γλώσσα: η C++ μεταγλωττίζεται απευθείας στον εγγενή κώδικα μιας μηχανής, επιτρέποντας της να είναι μια από τις γρηγορότερες γλώσσες προγραμματισμού στον κόσμο όταν είναι βελτιστοποιημένη,
- είναι μια μη ασφαλής γλώσσα δυνατών τύπων: η C++ είναι μια γλώσσα που υποθέτει ότι ο προγραμματιστής ξέρει τι κάνει, αλλά επιτρέπει πολύ μεγάλο ποσοστό ελέγχου σαν αποτέλεσμα,
- υποστηρίζει τόσο πρόδηλη όσο και τεκμαιρόμενη τυποποίηση(πληκτρολόγηση?), επιτρέποντας ελαστικότητα και ένα μέσο αποφυγής της πολυλογίας όταν αυτό επιθυμείται,
- υποστηρίζει τόσο στατικό όσο και δυναμικό έλεγχο τύπων: η C++ επιτρέπει τον έλεγχο των μετατροπών τύπων είτε κατά το χρόνο μεταγλώττισης είτε κατά το χρόνο εκτέλεσης, προσφέροντας ξανά άλλον ένα βαθμό ελαστικότητας (ο στατικός έλεγχος τύπων είναι παρ' όλα αυτά συχνότερος),
- προσφέρει πολλές επιλογές προτύπων: η C++ προσφέρει αξιοσημείωτη υποστήριξη για πρότυπα διαδικασιακού, γενικού και αντικειμενοστραφή προγραμματισμού, ενώ είναι δυνατά και πολλά περισσότερα πρότυπα,
- είναι φορητή: ως μια από τις γλώσσες προγραμματισμού που χρησιμοποιούνται συχνότερα σε όλο τον κόσμο και ως ανοιχτή γλώσσα, η C++ διαθέται μια μεγάλη εμβέλεια μεταγλωττιστών, οι οποίοι τρέχουν σε πολλές διαφορετικές πλατφόρμες που την υποστηρίζουν. Πηγαίος κώδικας που χρησιμοποιεί αποκλειστικά την κανονική βιβλιοθήκη της C++ θα τρέξει σε πολλές πλατφόρμες, με λίγες έως καθόλου αλλαγές,
- είναι συμβατή προς τα πάνω (upwards compatible) με τη C: η C++ όντας μια γλώσσα που κτίζεται άμεσα πάνω στη C, είναι συμβατή με σχεδόν όλο τον κώδικα της C. Η C++ μπορεί να χρησιμοποιήσει βιβλιοθήκες της C με λίγες έως καθόλου τροποποιήσεις του κώδικα των βιβλιοθηκών και
- έχει πολύ σημαντική υποστήριξη σε βιβλιοθήκες: μια αναζήτηση στη δημοφιλή ιστοσελίδα διαχείρισης προγραμμάτων SourceForge, θα αποφέρει πάνω από 3000 αποτελέσματα για βιβλιοθήκες C++.

Το αρχείο πηγαίου κώδικα της εφαρμογής dFrame είναι τύπου .cpp (C plus plus), δηλαδή αντιμετωπίζεται, μεταγλωττίζεται και εκτελείται ως μια εφαρμογή σε γλώσσα C++, αλλά είναι κατασκευάστηκε με βάση κυρίως εντολές της C, γεγονός το οποίο διευκόλυνε αρκετά η συμβατότητα προς τα πάνω (upwards compatibility) της C++, καθώς κατά κύριο λόγο η εφαρμογή χρησιμοποιεί βιβλιοθήκες της C. Παρ' όλα αυτα, η χρήση εντολών και συναρτήσεων της C++ ήταν απαραίτητη, ιδιαίτερα όταν γινόταν χρήση των εργαλείων και λειτουργιών που παρέχει η βιβλιοθήκη όρασης υπολογιστών OpenCV. Η επεξεργασία εικόνας και βίντεο, που αποτελεί το κύριο μέρος της εφαρμογής dFrame, είναι πολύ ευκολότερη και πιο αποδοτική με χρήση των λειτουργιών των βιβλιοθηκών της OpenCV, η οποία όμως είναι συμβατή με και παρέχει υποστήριξη σε προγράμματα ανεπτυγμένα σε

γλώσσα C++ και όχι C.

Μια εναλλακτική λύση για την κατασκευή της εφαρμογής εξαγωγής χαρακτηριστικών καρέ που αποτελεί το βασικό αντικείμενο της εργασίας αυτής, είναι η πλατφόρμα MATLAB. Το MATLAB χρησιμοποιείται ευρέως από μηχανικούς και επιστήμονες σε όλο τον κόσμο και έχει εφαρμογές σε μηχανική εκμάθηση (machine learning), επεξεργασία σήματος, επεξεργασία εικόνας, όραση υπολογιστών, τηλεπικοινωνίες, σχεδιασμό ελέγχου, ρομποτική και πολλά άλλα. Στο MATLAB μπορεί επίσης να ενσωματωθεί και η βιβλιοθήκη OpenCV που χρησιμοποιείται στην εφαρμογή dFrame. Η επιλογή της C++ έναντι του MATLAB για τη δημιουργία της εφαρμογής βασίστηκε κυρίως σε μεγαλύτερη προηγούμενη εξοικείωση με τη γλώσσα.

3.1.2 Eclipse IDE

Το πρόγραμμα Eclipse προσφέρει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (intergrated development environment, IDE) για μια μεγάλη ποικιλία γλωσσών και αρχιτεκτονικής προγραμματισμού. Το Eclipse είναι στο μεγαλύτερο μέρος του γραμμένο στη γλώσσα προγραμματισμού Java και είναι γνωστό κυρίως ως ολοκληρωμένο περιβάλλον ανάπτυξης στη γλώσσα Java αλλά προσφέρει επίσης τη δυνατότητα χρήσης των γλωσσών προγραμματισμού C και C++ που μας ενδιαφέρουν αλλά και των Ada, ABAP, COBOL, Fortran, Haskell, JavaScript, Perl, Prolog, Python, Ruby, Scala, Erlang και πολλών άλλων. Το Eclipse κυκλοφόρησε υπό τους όρους του Eclipse Public License και είναι ελεύθερο, ανοιχτού κώδικα λογισμικό.

Ιστορικά Στοιχεία

Το Eclipse βασίστηκε στη βασισμένη στο Smalltalk VisualAge οικογένεια προϊόντων ολοκληρωμένων περιβαλλόντων ανάπτυξης (IDE). Μια ομάδα κυρίως στο εργαστήριο IBM cary NC ανέπτυξε το νέο προϊόν ως ένα βασισμένο στη Java αντικαταστάτη των προϊόντων VisualAge, που καθιστούσαν δύσκολη την πρόσβαση σε επιμέρους classes. Το Νοέμβριο του 2001, ιδρύθηκε μια κοινοπραξία για την περεταίρω ανάπτυξη του Eclipse σαν λογισμικό ανοιχτού κώδικα. Υπολογίζεται ότι μέχρι τότε η IBM είχε ήδη επενδύσει περίπου 40 εκατομμύρια δολάρια. Τα αρχικά μέλη ήταν οι Borland, IBM, Merant, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft και WebGain. Ο αριθμός των μελών αυξήθηκε σε 80 μέχρι το τέλος του 2003 και τον Ιανουαριο του 2004 ιδρύθηκε το Ίδρυμα Eclipse (Eclipse Foundation).

Το Eclipse 3.0 (κυκλοφόρησε στις 21 Ιουνίου 2004) επέλεξε τις προδιαγραφές της πλατφόρμας υπηρεσιών OSGi σαν την αρχιτεκτονική χρόνου εκτέλεσης.

Αρχιτεκτονική

Το Eclipse περιέχει ένα βασικό χώρο εργασίας και ένα επεκτάσιμο σύστημα για plug-ins που επιτρέπει την τροποποίηση του περιβάλλοντος από το χρήστη. Χρησιμοποιεί τα plug-ins αυτά για την παροχή όλης της λειτουργικότητας εντός και πάνω στο σύστημα χρόνου εκτέλεσης. Το σύστημα χρόνου εκτέλεσης αυτό βασίζεται στο Equinox, το οποίο είναι μια μονάδα χρόνου εκτέλεσης που επιτρέπει στους προγραμματιστές να εκτελέσουν μια εφαρμογή ως ένα σύνολο από “πακέτα”, χρησιμοποιώντας την υποδομή κοινών υπηρεσιών.

Το πλαίσιο plug-ins επιτρέπει στην πλατφόρμα Eclipse, εκτός από την επέκτασή της χρησιμοποιώντας άλλες γλώσσες προγραμματισμού όπως η C και η Python, και το να δουλέψει με γλώσσες στοιχειοθεσίας όπως η LaTeX, εφαρμογές δικτύωσης όπως η telnet και με συστήματα διαχείρισης βάσεων δεδομένων. Η αρχιτεκτονική plug-in υποστηρίζει την πρόσθεση οποιασδήποτε

επιθυμητής επέκτασης στο περιβάλλον του Eclipse, όπως η διαχείριση διαμόρφωσης. Παρέχεται επίσης υποστήριξη Java και CVS στο Eclipse SDK, με plug-ins τρίτων να παρέχουν υποστήριξη για άλλα συστήματα ελέγχου.

Με την εξαίρεση ενός πυρήνα (kernel) μικρού χρόνου εκτέλεσης, τα πάντα στο Eclipse είναι plug-ins. Αυτό σημαίνει ότι κάθε νέο plug-in που αναπτύσσεται ενσωματώνεται στο Eclipse με ακριβώς τον ίδιο τρόπο με άλλα plug-ins. Από αυτή την πλευρά, όλες οι παροχές και τα χαρακτηριστικά είναι "ισότιμα δημιουργημένα". Το Eclipse παρέχει plug-ins για μια μεγάλη ποικιλία λειτουργιών, μερικά από τα οποία χρησιμοποιούν μέσω τρίτων τόσο ελεύθερα όσο και εμπορικά μοντέλα. Παραδείγματα plug-ins περιλαμβάνουν plug-ins για UML, για διαγράμματα ακολουθίας, για εξερεύνηση βάσεων δεδομένων και πολλά άλλα.

Το Eclipse SDK περιλαμβάνει τα εργαλεία ανάπτυξης Java Eclipse (Java Development Tools, JDT), προσφέροντας έτσι ένα ολοκληρωμένο περιβάλλον ανάπτυξης με ενσωματωμένο επαυξητικό (incremental) μεταγλωττιστή Java και ένα πλήρες μοντέλο των πηγαίων αρχείων της Java. Αυτό επιτρέπει τη χρήση προηγμένων τεχνικών τροποποίησης υπάρχοντος κώδικα αλλά και την ανάλυση κώδικα. Το ολοκληρωμένο περιβάλλον ανάπτυξης χρησιμοποιεί επίσης ένα χώρο εργασίας (workspace), στη συγκεκριμένη περίπτωση ένα σύνολο μετα-δεδομένων (metadata) πάνω σε επίπεδο χώρο αρχείων, επιτρέποντας εξωτερικές τροποποιήσεις αρχείων εφόσον στη συνέχεια ο αντίστοιχος "πόρος" του χώρου εργασίας ανανεώνεται.

Το Eclipse εφαρμόζει τα στοιχεία γραφικού ελέγχου της εργαλειοθήκης της Java που ονομάζεται SWT, ενώ οι περισσότερες εφαρμογές Java χρησιμοποιούν την πρότυπη εργαλειοθήκη "αφηρημένου παραθύρου", Abstract Window Toolkit (AWT) ή την Swing. Η επιφάνεια (interface) χρήστη του Eclipse χρησιμοποιεί επίσης ένα άμεσο επίπεδο γραφικού interface χρήστη που ονομάζεται Jface, το οποίο απλοποιεί την κατασκευή εφαρμογών βασισμένων στο SWT.

Πακέτα γλωσσών που αναπτύσσονται από το "Babel Project" παρέχουν μεταφράσεις σε πάνω από 12 φυσικές γλώσσες.

Πλατφόρμες Εφαρμογών

Το Eclipse παρέχει τις εξής πλατφόρμες:

-Πλατφόρμα Rich Client(RCP): χρησιμοποιείται για ανάπτυξη εφαρμογών γενικής χρήσης και αποτελείται από:

-το Equinox OSGi, ένα πρότυπο πλαίσιο ομαδοποίησης,

-την κεντρική πλατφόρμα, που εκκινεί το Eclipse και τρέχει τα plug-ins,

-τη Standard Widget Toolkit (SWT), μια φορητή widget εργαλειοθήκη,

-το JFace, κλάσεις θεατή που προσθέτουν προγραμματισμό model-view-controller (μοντέλο-όψη-χειριστής) στο SWT, buffers αρχείων, χειρισμό κειμένου, επεξεργαστές κειμένου και

-τον πάγκο εργασίας του Eclipse.

-Πλατφόρμα Διακομιστή: Το Eclipse υποστηρίζει ανάπτυξη εφαρμογών για τον Tomcat, τον GlassFish και πολλούς άλλους διακομιστές και συχνά μπορεί να εγκαταστήσει το συγκεκριμένο διακομιστή απευθείας από το ολοκληρωμένο περιβάλλον ανάπτυξης. Υποστηρίζει debugging από μακριά, επιτρέποντας στο χρήση να παρακολουθεί τις μεταβλητές και να εξετάσει τον κώδικα μιας εφαρμογής που τρέχει στον συνημμένο διακομιστή.

-Πλατφόρμα Εργαλείων Διαδικτύου: Η πλατφόρμα εργαλείων διαδικτύου (Web Tools Platform, WTP) είναι μια επέκταση της πλατφόρμας του Eclipse με εργαλεία για την ανάπτυξη εφαρμογών διαδικτύου και Java EE. Περιλαμβάνει γραφικούς επεξεργαστές και επεξεργαστές κώδικα για μια ποικιλία

γλωσσών, wizards και ενσωματωμένων εφαρμογών για την απλοποίηση της ανάπτυξης εφαρμογών, καθώς και εργαλεία και APIs για την υποστήριξη της ανάπτυξης, της λειτουργίας και του ελέγχου εφαρμογών.

-Πλατφόρμα Μοντελοποίησης: Το project μοντελοποίησης περιέχει όλα τα επίσημα προγράμματα του Eclipse Foundation που επικεντρώνονται σε τεχνολογίες ανάπτυξης βασισμένες σε μοντέλα. Όλα είναι συμβατά με το Πλαίσιο Μοντελοποίησης Eclipse που δημιουργήθηκε από την IBM. Τα προγράμματα αυτά χωρίζονται σε διάφορες κατηγορίες: Μεταμόρφωση Μοντέλου, Εργαλεία Ανάπτυξης Μοντέλου, Ανάπτυξη Συμπαγούς Συντακτικού, Τεχνολογία και Έρευνα, και Αμάλγαμα.

Το Eclipse υποστηρίζει μια πλούσια ποικιλία επεκτάσεων, προσθέτοντας υποστήριξη για Python μέσω του pydev, για ανάπτυξη εφαρμογών Android μέσω του ADT της Google, υποστήριξη για JavaFX μέσω e(fx)clipse, και πολλά άλλα, όπως και για JavaScript και jQuery. Το Valable είναι ένα plug-in του Eclipse για Vala.

Για την υλοποίηση της εφαρμογής dFrame έγινε χρήση του προγράμματος CDT της πλατφόρμας Eclipse.

Eclipse CDT (Εργαλεία Ανάπτυξης Εφαρμογών σε C/C++)

Το πρόγραμμα CDT (C/C++ Development Tooling) παρέχει ένα πλήρως λειτουργικό ολοκληρωμένο περιβάλλον ανάπτυξης εφαργμογών C και C++ βασισμένο στην πλατφόρμα Eclipse. Οι παροχές του περιλαμβάνουν: υποστήριξη για δημιουργία project και διαχείριση κατασκευής για διάφορες αλυσίδες εργαλείων, πρότυπη κατασκευή make, πλοήγηση στον κώδικα, διάφορα εργαλεία γνώσης πηγαίου κώδικα, όπως ιεραρχία τύπων, κλήση γραφήματος, περιήγηση includes, επεξεργαστή κώδικα με τονισμό σύνταξης, πλοήγηση αναδίπλωσης και υπερσυνδέσμων, μερική επαναδόμηση πηγαίου κώδικα και παραγωγή κώδικα, οπτικά εργαλεία debugging, όπως προγράμματα προβολής μνήμης, καταχωρητών και αποσυναρμολόγησης.

Το CDT είναι ένα ελεύθερο πρόγραμμα ανοιχτού κώδικα (σύμφωνα με την άδεια Common Public License), ανεπτυγμένο στην γλώσσα προγραμματισμού Java ως ένα σύνολο από plug-ins για την πλατφόρμα Eclipse SDK. Τα plug-ins αυτά προσθέτουν μια προοπτική C/C++ στον πάγκο εργασίας του Eclipse, που μπορεί πλέον να υποστηρίξει ανάπτυξη εφαρμογών σε C/C++ με κάποια views και wizards, καθώς και προηγμένη επεξεργασία και υποστήριξη debugging.

Λόγω της πολυπλοκότητας του, το CDT διαχωρίζεται σε διάφορα συστατικά τα οποία παίρνουν τη μορφή ξεχωριστών plug-ins. Κάθε συστατικό λειτουργεί σαν αυτόνομο πρόγραμμα, με το δικό του σύνολο από δεσμευτές, κατηγορίες bugs και λίστες μηνυμάτων. Παρ' όλα αυτά, απαιτούνται όλα τα plug-ins για τη σωστή λειτουργία του CDT. Παρατίθεται μια ολοκληρωμένη λίστα των plug-ins και συστατικών του CDT:

- το Κύριο CDT plugin είναι το “πλαίσιο” του CDT,
- το CDT Feature Eclipse είναι το συστατικό παροχών του CDT,
- το CDT Core, παρέχει το μοντέλο πυρήνα, το CDOM και τα βασικά συστατικά,
- το CDT UI είναι το UI(User Interface, διεπαφή χρήστη) πυρήνα, προβολείς, επεξεργαστές και wizards,
- το CDT Launch παρέχει τον μηχανισμό έναρξης για εξωτερικά εργαλεία όπως ο μεταγλωττιστής και ο debugger,
- το CDT Debug Core παρέχει λειτουργίες debugging,

- το CDT Debug UI παρέχει τη διεπαφή χρήστη για τους επεξεργαστές, προβολείς και wizards του debugging του CDT,
- το CDT Debug MI είναι η υποδοχή της εφαρμογής για debuggers συμβατούς με MI (Machine Interface, διεπαφή μηχανής).

Για να μπορέσει να λειτουργήσει το CDT, ο χρήστης πρέπει πριν την εγκατάστασή του να βεβαιωθεί ότι έχει εγκαταστήσει στον υπολογιστή του το μεταγλωττιστή C GNU (GCC) και όλα τα συνοδευτικά εργαλεία (make, binutils, GDB). Η εργασία αυτή και η εφαρμογή dFrame αναπτύχθηκαν σε λειτουργικό σύστημα Windows 10, όπου είχε εγκατασταθεί το Minimalist GNU for Windows (MinGW).

Το MinGW είναι μια συλλογή αρχείων κεφαλίδας (headers) ελεύθερα διαθέσιμων και ελεύθερης διανομής συγκεκριμένα για Windows και βιβλιοθήκες εισαγωγών (imports), συνδυασμένη με σύνολα εργαλείων GNU που επιτρέπουν στο χρήστη να παράγει εγγενή προγράμματα Windows που δε βασίζονται σε DLLs παρεχόμενα από τρίτους. Το MinGW είναι η καλύτερη επιλογή αν ο χρήστης θέλει να δημιουργήσει εφαρμογές συμβατές με POSIX.

Μια εναλλακτική λύση είναι η εγκατάσταση της εργαλειοθήκης Cygwin. Το Cygwin είναι ένα παρόμοιο με το UNIX περιβάλλον για Windows και περιλαμβάνει μια θύρα GCC, μαζί με όλα τα απαραίτητα εργαλεία ανάπτυξης εφαργμογών, τα οποία περιλαμβάνουν το automake και τον debugger GNU (GDB). Το Cygwin είναι κτισμένο γύρω από τη βιβλιοθήκη cygwin1.dll.

Σημαντικές Παροχές του CDT IDE

Το CDT IDE είναι κτισμένο γύρω από έναν επεκτάσιμο επεξεργαστή, ο οποίος παρέχεται από το CDT UI plug-in. Οι κύριες παροχές του CDT IDE είναι:

- Τονισμός Συντακτικού: το ολοκληρωμένο περιβάλλον ανάπτυξης CDT αναγνωρίζει το συντακτικό των γλωσσών C/C++ και παρέχει τονισμό του συντακτικού με πλήρως ρυθμιζόμενο χρωματισμό κώδικα και μορφοποίησή του.
- Σχεδιάγραμμα (outline): Η μονάδα παραθύρου σχεδιαγράμματος παρέχει μια γρήγορη απεικόνιση των διαδικασιών, μεταβλητών, δηλώσεων και συναρτήσεων που εμφανίζονται στον πηγαίο κώδικα. Με το σχεδιάγραμμα ο χρήστης μπορεί εύκολα να μεταβεί στην κατάλληλη αναφορά στον πηγαίο κώδικα ή ακόμα και να ψάξει τον πηγαίο κώδικα ολόκληρου του project.
- Code assist: Η παροχή αυτή ολοκλήρωσης κώδικα είναι παρόμοια με αυτές που βρίσκουμε στον Borland C++ Builder ή στο MS Visual Studio. Χρησιμοποιεί πρότυπα κώδικα και βοηθά απλά στην αποφυγή μικρών συντακτικών λαθών.
- Code Templates: Τα πρότυπα κώδικα, που χρησιμοποιούνται από την παροχή code assist, είναι ορισμοί κοινών συντακτικών κατασκευών των γλωσσών C/C++. Ο χρήστης μπορεί να ορίσει και τα δικά του πρότυπα κώδικα για την επέκταση των συντομεύσεων του, όπως αυτά για τις λέξεις-κλειδιά ημερομηνίας ή δημιουργού. Στην λίστα των προτύπων κώδικα στην εφαρμογή, ο χρήστης μπορεί να δει πλήρη κατάλογο των ήδη υπαρχόντων αλλά και να προσθέσει νέες. Τα πρότυπα κώδικα μπορούν επίσης να εξαχθούν και να εισαχθούν ως αρχεία XML.
- Code histoty (ιστορικό κώδικα): Ακόμα κι αν δε χρησιμοποιείται το CVS ή κάποιο άλλο λογισμικό χειρισμού έκδοσης πηγαίου κώδικα, ο χρήστης μπορεί να εντοπίσει τοπικές αλλαγές στον πηγαίο κώδικα ενός προγράμματος.

Για την υλοποίηση της εφαρμογής dFrame επιλέχθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse CDT για ανάπτυξη εφαρμογών στις γλώσσες προγραμματισμού C/C++ για πολλούς λόγους. Οι παροχές του CDT που αναφέρθηκαν παραπάνω παρέχουν σημαντικές διευκολύνσεις στην ανάπτυξη του πηγαίου κώδικα της εφαρμογής, ιδιαίτερα σε σχέση με τη χρήση απλού σημειωματάριου ή επεξεργαστή κειμένου σε συνδυασμό με τερματικό/command prompt. Χαρακτηριστικά του Eclipse CDT όπως ο τονισμός του συντακτικού και το σχεδιάγραμμα του προγράμματος βοηθούν σημαντικά τη δημιουργία και επεξεργασία του κώδικα, ενώ η ποικιλία επιλογών και λειτουργιών σχετικών με το debugging καθιστούν πολύ πιο εύκολο τον έλεγχο και τη διόρθωση τόσο σημαντικών λαθών στον πηγαίο κώδικα όσο και πιθανών παραλείψεων ή λανθασμένων χαρακτήρων που μπορεί να προέκυψαν από απροσεξία του χρήστη.

Η διεπαφή χρήστη του Eclipse CDT καθιστά ευκολότερη και την ενσωμάτωση στο ολοκληρωμένο περιβάλλον ανάπτυξης, ή απλά στο πρόγραμμα που μας ενδιαφέρει, άλλων συστατικών. Ένα παράδειγμα είναι και η βιβλιοθήκη OpenCV η οποία είναι απαραίτητη για την υλοποίηση του μεγαλύτερου τμήματος της εφαρμογής dFrame. Η βιβλιοθήκη αυτή είναι ελεύθερη και ανοικτού κώδικα, θα αναλυθεί στην επόμενη υποπαράγραφο και υλοποιεί εφαρμογές σχετικές με όραση υπολογιστών (Computer Vision). Χρησιμοποιώντας την θα πραγματοποιηθεί το κύριο και μεγαλύτερο κομμάτι της εφαρμογής dFrame, που αφορά την επεξεργασία των αρχείων βίντεο εισόδου, την εξαγωγή των καρέ που μας ενδιαφέρουν (επεξεργασία βίντεο και εικόνας) αλλά και την απεικόνιση των χαρακτηριστικών του προσώπου βάσει των κανονικοποιημένων συντεταγμένων τους στα καρέ αυτά. Η σύνδεση της OpenCV με το περιβάλλον Eclipse CDT απαιτεί τη χειροκίνητη ένταξη των απαραίτητων μονοπατιών μνήμης και διευθύνσεων αρχείων της OpenCV στα εργαλεία του CDT αλλά και τη σύνδεση με τα εργαλεία για το μεταγλωττιστή και το συνδέτη C/C++ με τις απαραίτητες βιβλιοθήκες.

Συμπερασματικά, η εργαλειοθήκη ανάπτυξης C/C++ (C/C++ Development Toolkit, CDT) παρέχει ένα δυνατό σύνολο plug-ins που μπορούν να βοηθήσουν στην ανάπτυξη εφαρμογών στις γλώσσες προγραμματισμού C/C++ με την πλατφόρμα Eclipse. Η χρήση του Eclipse CDT είναι πολύ πιο εύκολη και γρήγορη από ότι η δημιουργία της εφαρμογής σε επεξεργαστή κειμένου και η απευθείας μεταγλώττιση και εκτέλεσή της μέσω του MinGW στο τερματικό. Ανάλογα με την εξοικείωση του εκάστοτε χρήστη με το συγκεκριμένο περιβάλλον, εναλλακτικές επιλογές ολοκληρωμένων περιβαλλόντων για την ανάπτυξη εφαρμογών στις γλώσσες C/C++ περιλαμβάνουν το Microsoft Visual Studio, το οποίο όμως δεν είναι ελεύθερο λογισμικό, και το Netbeans.

3.1.3 Βιβλιοθήκη Όρασης Υπολογιστών Ανοιχτού Κώδικα OpenCV

Η OpenCV (Open source Computer Vision Library) είναι μια βιβλιοθήκη λογισμικού όρασης υπολογιστών και μηχανικής εκμάθησης (machine learning) ανοιχτού κώδικα. Η βιβλιοθήκη OpenCV κατασκευάστηκε για να παρέχει μια σταθερή υποδομή για εφαρμογές όρασης υπολογιστών αλλά και για να επιταχύνει τη χρήση μηχανικής αντίληψης στα διάφορα εμπορικά προϊόντα. Όντας προϊόν με άδεια BSD, η OpenCV διευκολύνει και τις επιχειρήσεις να χρησιμοποιήσουν και να τροποποιήσουν τον κώδικα.

Η βιβλιοθήκη περιέχει περισσότερους από 2500 βελτιστοποιημένους αλγορίθμους, οι οποίοι περιλαμβάνουν ένα περιεκτικό σύνολο τόσο κλασικών όσο και τελευταίας τεχνολογίας αλγορίθμων όρασης υπολογιστών και μηχανικής εκμάθησης. Οι αλγόριθμοι αυτοί μπορούν να χρησιμοποιηθούν για τον εντοπισμό και την αναγνώριση προσώπων, την αναγνώριση αντικειμένων, την ταξινόμηση των ανθρωπίνων ενεργειών σε αρχεία βίντεο, την ανίχνευση κινήσεων στην κάμερα, την ανίχνευση κινούμενων αντικειμένων, την εξαγωγή τρισδιάστατων μοντέλων από αντικείμενα, το συνδυασμό

εικόνων γιαι την παραγωγή μιας εικόνας υψηλής ευκρίνειας μιας ολόκληρης σκηνής, την εύρεση παρομοίων εικόνων από μια βάση δεδομένων με εικόνες, την αφαίρεση των κόκκινων ματιών από τις φωτογραφίες που λήφθηκαν με χρήση φλας, την παρακολούθηση κινήσεων των ματιών, την αναγνώριση σκηνικού και τον ορισμό δεικτών για την επικάλυψη με επαυξημένη πραγματικότητα (augmented reality) κ.α.

Η OpenCV έχει διεπαφές σε C++, C, Python, Java και MATLAB και υποστηρίζει τα λειτουργικά συστήματα Windows, Linux, Android και Mac OS. Η βιβλιοθήκη OpenCV τείνει κυρίως προς εφαρμογές όρασης υπολογιστών πραγματικού χρόνου και εκμεταλλεύεται τις οδηγίες MMX και SSE όταν είναι διαθέσιμες. Αναπτύσσονται επίσης πλήρεις διεπαφές CUDA και OpenCL. Υπάρχουν πάνω από 500 αλγόριθμοι και 5000 λειτουργίες που συνθέτουν ή υποστηρίζουν αυτούς τους αλγορίθμους. Η OpenCV είναι γραμμένη εγγενώς σε γλώσσα C++ και έχει μια πρότυπη διεπαφή που λειτουργεί απρόσκοπτα με δοχεία STL.

Η OpenCV έχει αρθρωτή δομή, γεγονός το οποίο σημαίνει ότι το πακέτο αυτό περιλαμβάνει διάφορες κοινόχρηστες ή και στατικές βιβλιοθήκες. Οι ενότητες που είναι διαθέσιμες περιγράφονται συνοπτικά στη συνέχεια:

- Λειτουργικότητα Πυρήνα: μια συμπαγής μονάδα που ορίζει βασικές λειτουργίες δεδομένων, συμπεριλαμβανομένου του πυκνού πολυδιάστατου πίνακα Mat και βασικών λειτουργιών που χρησιμοποιούνται από όλες τις άλλες ενότητες.
- Επεξεργασία Εικόνας: μια μονάδα επεξεργασίας εικόνας που περιλαμβάνει γραμμικό και μη γραμμικό φιλτράρισμα, γεωμετρικούς μετασχηματισμούς εικόνας (αλλαγή μεγέθους, στρέβλωση προοπτικής, γενικό remapping βασισμένο σε πίνακες), μετατροπή χρωματικού χώρου, ιστογράμματα κ.ο.κ.
- Video: μια μονάδα ανάλυσης βίντεο που περιλαμβάνει εκτίμηση κίνησης, αφαίρεση φόντου και αλγορίθμους εντοπισμού αντικειμένων.
- Calib3d: βασικοί αλγόριθμοι γεωμετρίας πολλαπλής όψης, διαμέτρηση μονής ή στέρεο κάμερας, εκτίμηση στάσης αντικειμένων, αλγόριθμοι στέρεο ανταπόκρισης και στοιχεία τρισδιάστατης αναδόμησης.
- Features2d: ανιχνευτές εξεχόντων χαρακτηριστικών, περιγραφείς και προσαρμοστές περιγραφέων.
- Objdetect: εντοπισμός αντικειμένων και στιγμιοτύπων των προκαθορισμένων κλάσεων (για παράδειγμα προσώπων, ματιών, ανθρώπων, αυτοκινήτων κ.ο.κ.).
- highgui: μια εύχρηστη διεπαφή για απλές δυνατότητες UI (διεπαφής χρήστη).
- Videoio: μια εύχρηστη διεπαφή για λήψη βίντεο και κωδικοποιητές βίντεο.
- GPU: αλγόριθμοι επιτάχυνσης GPU από διαφορετικές ενότητες OpenCV.
- Κάποιες άλλες βοηθητικές μονάδες, όπως περιτυλίγματα δοκιμών FLANN και Google, δέσμες Python και άλλα.

Στη συνέχεια παρουσιάζονται σύντομα κάποια στοιχεία διεπαφής προγραμματισμού εφαρμογών (Application Programming Interface, API) που χρησιμοποιούνται διεξοδικά στην OpenCV και η εξοικείωση με τα οποία θα διευκόλυνε το χρήστη.

Στοιχεία API

- cv Namespace: όλες οι κλάσεις και οι διαδικασίες της OpenCV τοποθετούνται στο cv namespace. Επομένως, για πρόσβαση σε αυτή τη λειτουργικότητα από τον πηγαίο κώδικα του χρήστη, απαιτείται η χρήση του προσδιοριστή cv:: ή της εντολής “using namespace cv;”
- Αυτόματη Διαχείριση Μνήμης: η OpenCV χειρίζεται όλη τη μνήμη αυτόματα. Πρώτα απ' όλα, οι std::vector, Mat και άλλες δομές δεδομένων που χρησιμοποιούνται από τις λειτουργίες και τις μεθόδους έχουν καταστροφείς που καταργούν τις εκχωρήσεις των buffers βασικής μνήμης όταν χρειάζεται. Αυτό σημαίνει ότι οι καταστροφείς δεν καταργούν πάντα τις εκχωρήσεις των buffers, όπως στην περίπτωση του Mat. Λαμβάνουν υπ' όψιν πιθανή κοινή χρήση δεδομένων. Ένας καταστροφέας μειώνει το μετρητή αναφοράς που σχετίζεται με τον buffer matrix δεδομένων. Κατάργηση της εκχώρησης του buffer γίνεται αν και μόνο αν ο μετρητής αναφοράς γίνει μηδέν, δηλαδή όταν δεν υπάρχουν άλλες δομές που να αναφέρονται στο συγκεκριμένο buffer. Παρομοίως, όταν αντιγράφεται ένα στιγμιότυπο Mat, δεν αντιγράφονται πραγματικά δεδομένα. Αντ' αυτού, ο μετρητής αναφοράς αυξάνεται για να απομνημονεύσει ότι υπάρχει άλλος ένας κάτοχος των ίδιων δεδομένων. Υπάρχει επίσης η μέθοδος Mat::clone που δημιουργεί ένα πλήρες αντίγραφο των δεδομένων matrix.
- Αυτόματη κατανομή των δεδομένων εξόδου: η OpenCV καταργεί τις εκχωρήσεις μνήμης αυτόματα, αλλά τις περισσότερες φορές κατανέμει αυτόματα τη μνήμη για τις παραμέτρους εξόδου συνάρτησης. Με αυτόν τον τρόπο, αν μια συνάρτηση έχει έναν ή περισσότερους πίνακες εισόδου (στιγμιότυπα cv::Mat) και κάποιους πίνακες εξόδου, οι πίνακες εξόδου κατανέμονται ή ανακατανέμονται αυτόματα. Το μέγεθος και ο τύπος των πινάκων εξόδου καθορίζονται από το μέγεθος και τον τύπο των πινάκων εισόδου. Αν χρειαστεί, οι συναρτήσεις δέχονται επιπλέον παραμέτρους, οι οποίες βοηθούν στην κατανόηση των ιδιοτήτων των πινάκων εξόδου. Το συστατικό-κλειδί αυτής της τεχνολογίας είναι η μέθοδος Mat::create. Δέχεται το επιθυμητό μέγεθος και τον επιθυμητό τύπο του πίνακα. Αν ο πίνακας έχει ήδη το καθορισμένο μέγεθος και τύπο, η μέθοδος δεν κάνει κάτι. Αλλιώς, ελευθερώνει τα προηγούμενα κατανεμημένα δεδομένα, αν υπάρχουν (αυτό το κομμάτι περιλαμβάνει μείωση του μετρητή αναφοράς και σύγκρισή του με το μηδέν), και στη συνέχεια κατανέμει ένα νέο buffer του απαιτούμενου μεγέθους. Οι περισσότερες συναρτήσεις καλούν τη μέθοδο Mat::create για κάθε πίνακα εξόδου, και έτσι υλοποιείται η αυτόματη κατανομή δεδομένων εξόδου.

  Αξιοσημείωτες εξαιρέσεις από αυτό το σχέδιο είναι οι cv::mixChannels, cv::RNG::fill και κάποιες άλλες συναρτήσεις και μέθοδοι. Δεν έχουν τη δυνατότητα κατανομής του πίνακα εξόδου, οπότε ο χρήστης θα πρέπει να το κάνει αυτό εκ των προτέρων.
- Αριθμητική Κορεσμού: Όντας βιβλιοθήκη όρασης υπολογιστών, η OpenCV ασχολείται αρκετά με εικονοστοιχεία μιας εικόνας, τα οποία είναι συχνά κωδικοποιημένα σε συμπαγή μορφή, 8- ή 16-bit ανά κανάλι, και έχουν επομένως περιορισμένη εμβέλεια τιμών. Επιπροσθέτως, κάποιες λειτουργίες σε εικόνες, όπως οι μετατροπές χρωματικού χώρου, οι προσαρμογές φωτεινότητας και αντίθεσης, η όξυνση, η σύνθετη παρεμβολή (δικυβική, Lanczos) μπορούν να παράγουν τιμές έξω από τη διαθέσιμη εμβέλεια. Αν απλά αποθηκευθούν τα χαμηλότερα 8(16) bits του αποτελέσματος, αυτό θα έχει ως αποτέλεσμα οπτικά αντικείμενα και μπορεί να επηρεάσει την περεταίρω ανάλυση εικόνας. Για να λυθεί αυτό το πρόβλημα χρησιμοποιείται η αποκαλούμενη Αριθμητική Κορεσμού. Για παράδειγμα, για την αποθήκευση του r, του αποτελέσματος μιας εργασίας, σε μια εικόνα 8-bit, βρίσκουμε την πλησιέστερη τιμή μέσα στο διάστημα 0...255:

  $$I(x,y)=\min(\max(\text{round}(r),0),255)$$

  Παρόμοιοι κανόνες εφαρμόζονται σε τύπους με πρόσημο 8-bit, 16-bit ή χωρίς πρόσημο. Αυτή η σημασιολογία εφαρμόζεται παντού στη βιβλιοθήκη. Σε κώδικα C++, πραγματοποείται με τη

χρήση των συναρτήσεων saturate_cast<>, οι οποίες είναι παρόμοιες με πρότυπες λειτουργίες cast της C++. Ο παραπάνω τύπος υλοποίεται ως εξής:

```
1 I.at<uchar>(y, x) = saturate_cast<uchar>(r);
```

όπου cv::uchar είναι ένας τύπος ακεραίου χωρίς πρόσημο 8-bit στην OpenCV. Στον βελτιστοποιημένο κώδικα SIMD, χρησιμοποιούνται τέτοιες εντολές SSE2 όπως οι paddusb, packuswb και άλλες. Οι εντολές αυτές συμβάλλουν στην επίτευξη ακριβώς της ίδιας συμπεριφοράς όπως στον κώδικα C++. Ο κορεσμός δεν εφαρμόζεται όταν το αποτέλεσμα είναι ακέραιος 32-bit.

– Σταθεροί Τύποι Εικονοστοιχείων, Περιορισμένη Χρήση Προτύπων: Τα πρότυπα είναι ένα πολύ χρήσιμο στοιχείο της C++, το οποίο καθιστά ικανή την υλοποίηση πολύ δυνατών, αποτελεσματικών αλλά παρ' όλα αυτά ασφαλών δομών δεδομένων και αλγορίθμων. Μολαταύτα, η εκτενής χρήση προτύπων μπορεί να αυξήσει δραματικά το χρόνο μεταγλώττισης και το μέγεθος του κώδικα. Είναι επίσης δύσκολο να διαχωριστούν μια διεπαφή και η υλοποίηση όταν χρησιμοποιούνται αποκλειστικά πρότυπα. Αυτό δε θα είχε μεγάλη σημασία για βασικούς, απλούς αλγορίθμους αλλά δε θα ήταν καλό για βιβλιοθήκες όρασης υπολογιστών όπου ένας μόνο αλγόριθμος μπορεί να αποτελείται από χιλιάδες γραμμές κώδικα. Για αυτό το λόγο, αλλά και για για την απλοποίηση της ανάπτυξης δεσμών για άλλες γλώσσες όπως οι Python, Java, MATLAB, οι οποίες δεν έχουν κανένα πρότυπο ή έχουν περιορισμένες δυνατότητες προτύπων, η τρέχουσα υλοποίηση της OpenCV βασίζεται στον πολυμορφισμό και σε αποστολές χρόνου εκτέλεσης αντί για πρότυπα. Στα σημεία όπου οι αποστολές χρόνου εκτέλεσης (runtime dispatching) θα ήταν πολύ αργές (όπως οι φορείς πρόσβασης εικονοστοιχείων), αδύνατες (γενική υλοποίηση Ptr<>), ή απλά πολύ άβολες (saturate_cast<>()), η τρέχουσα υλοποίηση εισάγει μικρές πρότυπες κλάσεις, μεθόδους και συναρτήσεις. Σε οποιοδήποτε άλλο σημείο της τρέχουσας έκδοσης της OpenCV η χρήση προτύπων είναι περιορισμένη.
Συνεπώς, το σταθερό σύνολο αρχικών τύπων δεδομένων πάνω στο οποίο μπορεί να λειτουργήσει η βιβλιοθήκη είναι περιορισμένο. Αυτό σημαίνει ότι τα στοιχεία των πινάκων θα πρέπει να είναι ενός από τους ακόλουθους τύπους:

- 8-bit ακέραιος χωρίς πρόσημο (uchar)
- 8-bit ακέραιος με πρόσημο (schar)
- 16-bit ακέραιος χωρίς πρόσημο (ushort)
- 16-bit ακέραιος με πρόσημο (short)
- 32-bit ακέραιος με πρόσημο (int)
- 32-bit αριθμός κινητής υποδιαστολής (float)
- 64-bit αριθμός κινητής υποδιαστολής (double)
- μια πλειάδα διάφορων στοιχείων, όπου όλα τα στεοιχεία έχουν τον ίδιο τύπο (έναν από τους παραπάνω). Ένας πίνακας του οποίου τα στοιχεία είναι τέτοιες πλειάδες, αποκαλείται πίνακας πολλαπλών καναλιών (multi-channel), σε αντίθεση με τους πίνακες μονού καναλιού, των οποίων τα στοιχεία είναι βαθμωτές τιμές. Ο μέγιστος πιθανός αριθμός καναλιών καθορίζεται από τη σταθερά CV_CN_MAX, η οποία έχει οριστεί ως 512.

Για αυτούς τους βασικούς τύπους, εφαρμόζεται η ακόλουθη αρίθμηση:

```
enum { CV_8U=0, CV_8S=1, CV_16U=2, CV_16S=3, CV_32S=4, CV_32F=5,
CV_64F=6 };
```

Τύποι πολλαπλών καναλιών (n-channel) μπορούν να διευκρινιστούν χρησιμοποιώντας τις ακόλουθες επιλογές:

- CV_8UC1 ... CV_64FC4 σταθερές (για αριθμό καναλιών από 1 έως 4)
- CV_8UC(n) … CV_64FC(n) ή CV_MAKETYPE(CV_8U, n) … CV_MAKETYPE(CV_64F, n) macros όταν ο αριθμός των καναλιών είναι μεγαλύτερος του 4 ή άγνωστος τη στιγμή της μεταγλώττισης.

Πίνακες με πιο πολύπλοκα στοιχεία δεν μπορούν να κατασκευαστούν ή να επεξεργαστούν από την OpenCV. Επιπροσθέτως, κάθε συνάρτηση ή μέθοδος μπορεί να χειριστεί μόνο ένα υποσύνολο όλων των δυνατών τύπων πινάκων. Συνήθως όσο πιο πολύπλοκος είναι ο αλγόριθμος, τόσο μικρότερο είναι το υποσύνολο των μορφών. Ακολουθούν κοινά παραδείγματα τέτοιων περιορισμών:

- Ο αλγόριθμος ανίχνευσης προσώπου λειτουργεί μόνο με ασπρόμαυρες ή έγχρωμες εικόνες 8-bit.
- Οι συναρτήσεις γραμμικής άλγεβρας και οι περισσότεροι αλγόριθμοι μηχανικής εκμάθησης λειτουργούν μόνο με πίνακες κινητής υποδιαστολής.
- Βασικές συναρτήσεις, όπως η cv::add, υποστηρίζουν όλους τους τύπους.
- Οι συναρτήσεις μετατροπής χρωματικού χώρου υποστηρίζουν τύπους 8-bit χωρίς πρόσημο, 16-bit χωρίς πρόσημο και 32-bit κινητής υποδιαστολής.

– Πίνακας Εισόδου και Πίνακας Εξόδου (InputArray και OutputArray): πολλές συναρτήσεις της OpenCV επεξεργάζονται διδιάστατους ή πολυδιάστατους αριθμητικούς πίνακες. Τέτοιες συναρτήσεις συνήθως δέχονται τα cppMat ως παραμέτρους, αλλά σε κάποιες περιπτώσεις είναι ευκολότερη η χρήση του std::vector<> (για παράδειγμα, σε ένα σύνολο σημείων) ή το Matx<> (π.χ. για μήτρες ομογραφίας 3x3). Για την αποφυγή πολλών αντιγράφων στο API, έχουν εισαχθεί ειδικές "πληρεξούσιες" κλάσεις. Η βασική "πληρεξούσια" κλάση είναι ο InputArray. Χρησιμοποιείται για προσπέλαση πινάκων read-only (μόνο ανάγνωσης) στην είσοδο μιας συνάρτησης. Η κλάση OutputArray, η οποία προέρχεται από την InputArray, χρησιμοποιείται για να καθορίσει έναν πίνακα εξόδου για μια συνάρτηση. Μπορούμε να υποθέσουμε ότι μπορεί πάντα να γίνει χρήσω των Mat, std::vector<>, Matx<>, Vec<> ή Scalar αντί των κλάσεων Input/OutputArray. Όταν μια συνάρτηση έχει προαιρετική είσοδο ή έξοδο πίνακα και δεν έχουμε ή δε θέλουμε να χρησιμοποιήσουμε έναν πίνακα, χρησιμοποιείται η cv::noArray().

– Χειρισμός Λαθών: η OpenCV χρησιμοποιεί εξαιρέσεις για να σηματοδοτήσει κρίσιμα λάθη. Όταν τα δεδομένα εισόδου έχουν μια σωστή μορφή και ανήκουν στην καθορισμένη εμβέλεια τιμων, αλλά ο αλγόριθμος δεν μπορεί τα εκτελεστεί επιτυχώς για κάποιο λόγο (για παράδειγμα, ο αλγόριθμος βελτιστοποίησης δεν συνέκλινε), επιστρέφει έναν ειδικό κωδικό λάθους (συνήθως μια boolean μεταβλητή.

Οι εξαιρέσεις μπορούν να είναι στιγμιότυπα της κλάσης cv::Exception ή των παραγώγων αυτής. Με τη σειρά της η cv::Exception είναι παράγωγος της std::exception. Μπορεί με αυτόν τον τρόπο να αντιμετωπιστεί στον κώδικα χρησιμοποιώντας άλλα συστατικά πρότυπων βιβλιοθηκών της C++. Η εξαίρεση συνήθως γίνεται χρησιμοποιώντας είτε την μακροεντολή CV_Error(errcode, description), ή την παρόμοια με την printf παραλλαγή της, CV_Error_(errcode, printf-spec, (printf-args)) ή χρησιμοποιώντας την μακροεντολή CV_Assert(condition) που ελέγχει τη συνθήκη (condition) και εμφανίζει εξαίρεση όταν αυτή δεν ικανοποιείται. Για κώδικα κρίσιμης απόδοσης, υπάρχει η CV_DbgAssert(condition) η οποία συγκρατείται μόνο στη διαμόρφωση Debug. Εξαιτίας της αυτόματης διαχείρισης μνήμης, καταργούνται αυτόματα οι εκχωρήσεις όλων των άμεσων buffers σε περίπτωση κάποιου ξαφνικού λάθους. Χρειάζεται μόνο η πρόσθεση μιας εντολής try για τη σύλληψη των εξαιρέσεων, αν αυτό απαιτείται.

– Multi-threading και Re-enterability: η τρέχουσα υλοποίηση της OpenCV είναι πλήρως re-enterable. Αυτό σημαίνει ότι η ίδια συνάρτηση, η ίδια constant μέθοδος στιγμιοτύπου κλάσης ή η ίδια non-constant μέθοδος διαφορετικών στιγμιοτύπων κλάσεων μπορούν να κληθούν απο διαφορετικά threads (νήματα). Επίσης η cv::Mat μπορεί να χρησιμοποιηθεί σε διαφορετικά threads επειδή οι λειτουργίες μετρήσεως αναφορών χρησιμοποιούν τις ειδικές για την αρχιτεκτονική ατομικές οδηγίες.

Στην υλοποίηση της εφαρμογής dFrame χρησιμοποιήθηκαν διάφορες συναρτήσεις και μέθοδοι από διαφορετικές βιβλιοθήκες της OpenCV, οι οποίες αφορούσαν κυρίως την ανάλυση βίντεο και εικόνων, την ανίχνευση αντικειμένων και χαρακτηριστικών, την εκτίμηση της κίνησης και την διεπαφή χρήστη. Τα στοιχεία της OpenCV που χρησιμοποιήθηκαν παρουσιάζονται πιο αναλυτικά στο επόμενο κεφάλαιο, μαζί με τον αλγόριθμο της dFrame.

3.2 Δεδομένα Εισόδου Εφαρμογής dFrame

Στην υποπαράγραφο αυτή παρουσιάζονται συνοπτικά τα δεδομένα εισόδου και ενέργειες που ήταν απαραίτητες για την πραγματοποίηση και την εκτέλεση της εφαρμογής dFrame για εξαγωγή χαρακτηριστικών καρέ από ακολουθίες βίντεο παικτών σε συνεδρίες του βιντεοπαιχνιδιού Super Mario Bros.

Η εφαρομογή dFrame δέχεται ως είσοδο ένα αρχείο βίντεο (τύπου .avi) με τις αντιδράσεις ενός παίκτη κατά τη διάρκεια συνεδρίας παιχνιδιού της ελεύθερης παραλλαγής ανοικτού κώδικα του Super Mario Bros, Infinite Mario Bros. Το βίντεο αυτό συνοδεύεται από το αντίστοιχο αρχείο κειμένου, που περιέχει τις μετρήσεις των συντεταγμένων των χαρακτηριστικών του προσώπου του παίκτη σε διάφορες χρονικές στιγμές, όπως αυτό εξάγεται από την εκτέλεση της εφαρμογής FacialFeat που εξετάστηκε στο προηγούμενο Κεφάλαιο. Πιο συγκεκριμένα, από την εφαρμογή FacialFeat παράγεται για το αρχείο βίντεο που εξετάζουμε ένα αρχείο κειμένου (τύπου .txt) που περιέχει κανονικοποιημένες συντεταγμένες για την απεικόνιση χαρακτηριστικών του προσώπου του παίκτη κάθε χρονική στιγμή ως εξής:

- τρία σημεία για το δεξί μάτι (αριστερή γωνία, κέντρο, δεξιά γωνία),
- τρία σημεία για το αριστερό μάτι (αριστερή γωνία, κέντρο, δεξιά γωνία),
- δύο σημεία για τις βλεφαρίδες του δεξιού ματιού (πάνω και κάτω βλεφαρίδα),
- δύο σημεία για τις βλεφαρίδες του αριστερού ματιού (πάνω και κάτω βλεφαρίδα),
- τέσσερα σημεία για το στόμα (αριστερή και δεξιά γωνία του στόματος, ανώτερο και κατώτερο σημείο του στόματος),
- ένα σημείο για τη μύτη (άκρη της μύτης),
- δύο σημεία για το δεξί φρύδι (εξωτερική και εσωτερική γωνία του φρυδιού) και
- δύο σημεία για το αριστερό φρύδι (εξωτερική και εσωτερική γωνία του φρυδιού).

Οι τιμές της πρώτης στήλης κάθε τέτοιου αρχείου κειμένου αντιπροσωπεύουν τα milliseconds που αντιστοιχούν σε κάθε καρέ του αρχείου βίντεο εισόδου, ενώ η γραμμή που ακολουθεί κάθε τέτοια τιμή, περιέχει τις τιμές των κανονικοποιημένων συντεταγμένων αυτών των 19 σημείων του προσώπου

για το καρέ αυτό.

Σκοπός της εφαρμογής dFrame, όπως έχουμε αναφέρει, είναι η εξαγωγή από αρχείο βίντεο με τις αντιδράσεις ενός παίκτη, των καρέ αυτών για τα οποία παρουσιάζονται οι μεγαλύτερες αποκλίσεις συντεταγμένων των 19 σημείων του προσώπου σε σχέση με κάποια σημεία αναφοράς που ορίζουμε για κάθε παίκτη. Η έξοδος αυτή της εφαρμογής εξαρτάται από την επεξεργασία των δύο αρχείων εισόδου (βίντεο και αντίστοιχο αρχείο κειμένου), η οποία επιτυγχάνεται με χρήση των κατάλληλων εργαλείων και βιβλιοθηκών της OpenCV. Τα αρχεία αυτά, τα οποία θα πρέπει να βρίσκονται στο χώρο εργασίας (workspace) του περιβάλλοντος Eclipse CDT που έχει δημιουργηθεί για το dFrame C++ project, δίνονται από τον χρήστη σαν παράμετροι(arguments) κατά την εκτέλεση της εφαρμογής. Αν επιθυμεί ο χρήστης μπορει να δώσει τα ονόματα των αρχείων αυτών μαζί με ένα όνομα αρχείου κειμένου για τη δημιουργία νέου αρχείου κειμένου εξόδου που θα περιέχει τις τιμές των συντεταγμένων των καρέ που αναζητούμε και τα αντίστοιχα milliseconds.

Μετά την εύρεση των τιμών milliseconds και συντεταγμένων που αντιστοιχούν στα καρέ μεγαλύτερης απόκλισης, η εφαρμογή πρέπει να εντοπίσει τα καρέ αυτά στο αρχείο βίντεο εισόδου και στη συνέχεια να εξάγει τα συγκεκριμένα στιγμιότυπα ως αρχεία εικόνας. Στα αρχεία εικόνας που παράγονται, η εφαργμογή dFrame απεικονίζει τα 19 σημεία του προσώπου του παίκτη, βάσει των συντεταγμένων που περιέχονται στο αρχείο κειμένου εισόδου. Ο εντοπισμός και η εξαγωγή των χαρακτηριστικών καρέ, καθώς και η απεικόνιση των χαρακτηριστικών του προσώπου του παίκτη (τα οποία εμφανίζονται με τη μορφή μικρών κύκλων ρυθμιζομένου χρώματος και μεγέθους) πάνω στα αρχεία εικόνας που προκύπτουν γίνεται με εισαγωγή βιβλιοθηκών και χρήση λειτουργιών της OpenCV. Η OpenCV διευκολύνει επίσης την προσπέλαση του αρχείου βίντεο εισόδου και την αποθήκευσή του σε μεταβλητή για επεξεργασία.

Τα στοιχεία αυτά της OpenCV που χρησιμοποιήθηκαν αναλύονται περισσότερο στο επόμενο κεφάλαιο, όπου παρουσιάζεται εκτενώς ο αλγόριθμος βάσει του οποίου κατασκευάστηκε η dFrame και εξετάζονται περεταίρω οι εντολές, συναρτήσεις και διαδικασίες της γλώσσας προγραμματισμού C++ που την αποτελούν.

4. Η Εφαρμογή dFrame

Στο κεφάλαιο αυτό παρουσιάζεται το κεντρικό αντικείμενο της διπλωματικής εργασίας αυτής, η εφαρμογή dFrame και ο αλγόριθμος στον οποίο βασίζεται η υλοποίησή της. Η εφαρμογή αναπτύχθηκε στη γλώσσα προγραμματισμού C++, με κάποιες εντολές και συναρτήσεις της C και της βιβλιοθήκης όρασης υπολογιστών ανοιχτού κώδικα OpenCV. Η υλοποίηση έγινε στο ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse IDE, και συγκεκριμένα με χρήση του Eclipse CDT (C/C++ Development Toolkit), μιας εργαλειοθήκης της πλατφόρμας Eclipse για ανάπτυξη εφαρμογών στις γλώσσες C και C++. Η εφαρμογή εκτελείται σε λειτουργικό σύστημα Windows 10 και ο μεταγλωττιστής με τον οποίο συνδέθηκε η πλατφόρμα Eclipse βασίζεται στο σύστημα MinGW (Minimalist GNU for Windows), το οποίο χρησιμοποιεί τον μεταγλωττιστή C GNU (GCC) για την παραγωγή προγραμμάτων C/C++ σε Windows.

Το πρόγραμμα σε C++ εκτελείται από το Eclipse αφού του δώσουμε ως παραμέτρους στην είσοδο τα ονόματα δύο αρχείων εισόδου. Το πρώτο αρχείο εισόδου είναι ένα αρχείο βίντεο τύπου .avi που περιέχει τη συνεδρία παιχνιδιού Super Mario Bros ενός παίκτη. Το δεύτερο αρχείο εισόδου είναι ένα αρχείο κειμένου (τύπου .txt), το οποίο περιέχει τις τιμές των κανονικοποιημένων συντεταγμένων 19 σημείων του προσώπου του παίκτη, οι οποίες λαμβάνονται για κάθε καρέ της ακολουθίας βίντεο, καθώς και μια τιμή για τη χρονική στιγμή που αντιστοιχεί σε καθένα από αυτά τα καρέ, σε milliseconds. Αυτά τα αρχεία κειμένου είναι τα αποτελέσματα της εκτέλεσης της εφαρμογής FacialFeat για εντοπισμό των χαρακτηριστικών του προσώπου του παίκτη, όπως παρουσιάστηκε σε προηγούμενο κεφάλαιο.

Η εφαρμογή dFrame, προσπελαύνοντας το αρχείο κειμένου εισόδου, δημιουργεί δύο πίνακες (έναν μονοδιάστατο κι έναν διδιάστατο) στους οποίους αποθηκεύει τον μετρητή των milliseconds και τις τιμές των συντεταγμένων των 19 σημείων των χαρακτηριστικών προσώπου του παίκτη. Μέσα στο πρόγραμμα έχει γίνει επιλογή ενός συνόλου από τιμές αναφοράς για τις συντεταγμένες καθενός από αυτά τα 19 σημεία, και αποθήκευσή του σε έναν (μονοδιάστατο) πίνακα. Με τα περιεχόμενα του πίνακα αυτού, συγκρίνονται οι αντίστοιχες τιμές των συντεταγμένων του πίνακα που δημιουργήθηκε με βάση τα δεδομένα του αρχείου κειμένου εισόδου, και συγκρατούνται οι τιμές που είχαν απόκλιση από τα σημεία αναφοράς μεγαλύτερη του 10%, καθώς και οι τιμές των milliseconds στων οποίων τα καρέ αντιστοιχούν οι αποκλίσεις που εντοπίστηκαν. Εάν ο χρήστης επιθυμεί, δίνοντας ως παράμετρο στην εφαρμογή (μέσω του Eclipse) ένα νέο όνομα αρχείου κειμένου, μπορεί να εξάγει στο καινούριο αρχείο κειμένου που δημιουργείται από την εκτέλεση της dFrame τις τιμές των milliseconds και των κανονικοποιημένων συντεταγμένων των καρέ στα οποία είχαμε τις μεγαλύτερες του 10% αποκλίσεις που αναζητούνται.

Σαν είσοδος στην εφαρμογή dFrame δίνεται και το αρχείο βίντεο από το οποίο προέκυψε (με εκτέλεση της εφαρμογής FacialFeat) το αρχείο κειμένου εισόδου. Δημιουργείται ένας νέος μονοδιάστατος πίνακας, ο οποίος περιέχει τιμές των χρονικών στιγμών σε milliseconds και προκύπτει από τη διαδικασία εύρεσης των σημείων των οποίων οι κανονικοποιημένες συντεταγμένες παρουσιάζουν μεγαλύτερη του 10% απόκλιση από τις επιλεγμένες τιμές αναφοράς. Με βάση τον πίνακα αυτόν, θα πραγματοποιηθεί η εύρεση και εξαγωγή των χαρακτηριστικών καρέ από το αρχείο βίντεο. Χρησιμοποιώντας τα εργαλεία και τις βιβλιοθήκες της OpenCV, το πρόγραμμα προσπελαύνει το αρχείο βίντεο εισόδου και, λαμβάνοντας τις τιμές των milliseconds του πίνακα, ενοπίζει και εξάγει τα αντίστοιχα καρέ ως αρχεία εικόνας τύπου .jpg. Σκοπός του προγράμματός μας είναι και η

απεικόνιση των 19 σημείων αυτών σε καθένα από τα αρχεία εικόνας εξόδου. Αυτό επιτυγχάνεται με τη χρήση μεθόδων της OpenCV, βάσει των δεδομένων που λαμβάνουμε από το νέο διδιάστατο πίνακα στον οποίο αποθηκεύθηκαν οι συντεταγμένες των σημείων που μας ενδιαφέρουν για κάθε χαρακτηριστικό καρέ.

Στη συνέχεια παρουσιάζεται αναλυτικά η εφαρμογή dFrame, ο αλγόριθμος και ο πηγαίος κώδικας που την υλοποιούν. Εξετάζονται επίσης οι συναρτήσεις των γλωσσών C/C++ και οι μέθοδοι και βιβλιοθήκες της OpenCV που χρησιμοποιήθηκαν, αλλά και ο τρόπος εκτέλεσης της εφαρμογής και τα τελικά αποτελέσματά της.

4.1 Προετοιμασία για την Υλοποίηση και την Εκτέλεση της Εφαρμογής

Παρουσιάζονται συνοπτικά οι διαδικασίες που ακολουθήθηκαν για την υλοποίηση της εφαρμογής dFrame στο ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse με χρήση της εργαλειοθήκης ανάπτυξης εφραμογών σε C/C++, καθώς και για την εκτέλεσή της. Θα εξεταστούν επίσης και οι βιβλιοθήκες όρασης υπολογιστών της OpenCV που απαιτούνται για τη σωστή εκτέλεση του προγράμματος και τις οποίες συνδέσαμε με το σύστημα μεταγλωττιστή GNU της C MinGW για Windows καθώς και με το Eclipse CDT.

4.1.1 Προετοιμασία του Eclipse IDE

Για την υλοποίηση της εφαρμογής dFrame εγκαταστάθηκε από την ιστοσελίδα http://www.eclipse.org/downloads/, σε λειτουργικό σύστημα Windows 10, η έκδοση Neon (4.6) του Eclipse IDE για προγραμματιστές C/C++, το οποίο συμπεριλαμβάνει την εργαλειοθήκη για ανάπτυξη εφαρμογών σε C/C++, CDT.

Για την υλοποίηση του προγράμματος, δημιουργήθηκε ένα νέο C/C++ project στο χώρο εργασίας (workspace) του Eclipse IDE, για το οποίο στη συνέχεια επιλέγεται ο μεταγλωττιστής MinGW. Το μονοπάτι των αρχείων του συστήματος μεταγλώττισης MinGW είχε ήδη προστεθεί στις τιμές της μεταβλητής PATH του χρήστη του υπολογιστή,, για τη σωστή εκτέλεση εφαρμογών στις γλώσσες C και C++.

Μεταβλητή Περιβάλλοντος PATH

Η PATH είναι μια μεταβλητή περιβάλλοντος που εμφανίζεται σε διάφορα λειτουργικά συστήματα, όπως συστήματα παρόμοια με Unix, DOS OS και Microsoft Windows. Καθορίζει ένα σύνολο μονοπατιών (directories) όπου βρίσκονται εκτελέσιμα προγράμματα (τύπου EXE ή COM). Γενικά, κάθε εκτελούμενη διαδικασία ή συνεδρία χρήστη έχει τη δική της ρύθμιση PATH. Στα λειτουργικά συστήματα DOS, OS/2 και Windows η μεταβλητή %PATH% ορίζεται ως μια λίστα ενός ή περισσοτέρων ονομάτων μονοπατιών, διαχωρισμένων με χαρακτήρες ελληνικού ερωτηματικού (άνω τελείας) (;). Πολλά προγράμματα δεν εμφανίζονται στην PATH καθώς δεν είναι σχεδιασμένα για να εκτελούνται από παράθυρο εντολών (τερματικό) αλλά από γραφική διεπαφή χρήστη (GUI). Μερικά προγράμματα ίσως προσθέσουν το μονοπάτι τους στην αρχή του περιεχομένου της μεταβλητής PATH κατά την εγκατάσταση, για να επιταχύνουν τη διαδικασία αναζήτησης ή/και να παρακάμψουν εντολές OS.

Η μεταβλητή PATH διευκολύνει την εκτέλεση συχνά χρησιμοποιουμένων προγράμματων που

βρίσκονται σε δικούς τους φακέλους. Παρ' όλα αυτά, αν δε χρησιμοποιηθεί σωστά, η τιμή της μεταβλητής PATH μπορεί να επιβραδύνει αρκετά το λειτουργικό σύστημα, κάνοντας αναζήτηση σε υπερβολικά πολλές ή λανθασμένες τοποθεσίες. Οι λανθασμένες τοποθεσίες μπορούν επίσης να σταματήσουν εντελώς την εκτέλεση διάφορων υπηρεσιών, ειδικά της υπηρεσίας “Server”(διακομιστή), η οποία συνήθως είναι ένα εξάρτημα που χρησιμοποιείται για άλλες υπηρεσίες μέσα σε περιβάλλον διακομιστή Windows.

Απαραίτητη για τη σωστή υλοποίηση και εκτέλεση του project dFrame στο ολοκληρωμένο περιβάλλον ανάπτυξης του Eclipse είναι και η σύνδεση του συστήματος MinGW με τις βιβλιοθήκες που χρειαζόμαστε από την OpenCV και με το μονοπάτι αυτής στον υπολογιστή. Συνδέσαμε το μονοπάτι που είχε δημιουργηθεί για χρήση των εργαλείων της OpenCV από το MinGW, μέσω της καρτέλας libraries στις ιδιότητες του C++ Linker του MinGW, με την επιλογή -L(Library search path, μονοπάτι αναζήτησης βιβλιοθήκης) “C:\opencv\MinGW\bin”. Προσθέσαμε επίσης στην ίδια καρτέλα, με την επιλογή -l (libraries), τις βιβλιοθήκες της OpenCV που χρειαζόμαστε. Στις ιδιότητες του μεταγλωττιστή C++ του συστήματος MinGW και συγκεκριμένα στην καρτέλα includes, προσθέτουμε με την επιλογή -I(Include path) “C:\opencv\build\include”, το μονοπάτι που περιέχει τις κεφαλίδες που θέλουμε να εισάγουμε στο πρόγραμμα.

4.1.2 Βιβλιοθήκες Όρασης Υπολογιστών της OpenCV

Ο C++ Linker του συστήματος MinGW στο Eclipse CDT συνδέει με την επιλογή -l, κατά τη διάρκεια της μεταγλώττισης/σύνδεσης του προγράμματος dFrame, τις εξής βιβλιοθήκες όρασης υπολογιστών της OpenCV (έκδοση 3.0.0):

- calib3d (Camera Calibration and 3D Reconstruction, libopencv_calib3d300): Βαθμονόμηση κάμερας και τρισδιάστατη ανακατασκευή. Αυτό το κομμάτι περιέχει συναρτήσεις που χρησιμοποιούν ένα pinhole μοντέλο κάμερας. Στο μοντέλο αυτό, μια προβολή σκηνής σχηματίζεται προβάλλοντας 3D σημεία στο επίπεδο της εικόνας χρησιμοποιώντας έναν μετασχηματισμό προοπτικής.
- core (libopencv_core300): Λειτουργικότητα Πυρήνα. Το κομμάτι αυτό περιέχει τις βασικές δομές που χρησιμοποιεί η OpenCV, βασικές δομές και λειτουργίες της C, δυναμικές δομές, λειτουργίες σε πίνακες, συναρτήσεις σχεδιασμού, ομαδοποίηση, συναρτήσεις και μακροεντολές συστήματος και διαλειτουργικότητα OpenGL.
- flann (Fast Library for Approximate Nearest Neighbors , libopencv_flann300): Γρήγορη Βιβλιοθήκη για Προσέγγιση των Πλησιέστερων Γειτόνων. Το κομμάτι αυτό τεκμηριώνει τη διεπαφή της OpenCV για τη βιβλιοθήκη FLANN. Η FLANN είναι μια βιβλιοθήκη που περιέχει μια συλλογή από αλγορίθμους βελτιστοποιημένους για τη γρήγορη αναζήτηση του πλησιέστερου γείτονα σε μεγάλα σύνολα δεδομένων και για χαρακτηριστικά υψηλών διαστάσεων.
- highgui (High-level GUI and Media I/O, libopencv_highgui300): Γραφική Διεπαφή Χρήστη (GUI) Υψηλού Επιπέδου και Είσοδος/Εξοδων Πολυμέσων. Ενώ η OpenCV σχεδιάστηκε για χρήση σε εφαρμογές πλήρους κλίμακας και μπορεί να χρησιμοποιηθεί μέσα σε λειτουργικά πλούσια UI πλαίσια (όπως τα Qt, WinForms ή Cocoa) ή και χωρίς καμία διεπαφή χρήστη (UI), κάποιες φορές απαιτείται γρήγορη δοκιμή λειτουργικότητας και απεικόνιση των αποτελεσμάτων. Για αυτό το λόγο σχεδιάστηκε η μονάδα highgui. Παρέχει έυκολη διεπαφή για:
    - τη δημιουργία και το χειρισμό παραθύρων που μπορούν να εμφανίσουν εικόνες και να “θυμούνται” το περιεχόμενό τους (δε χρειάζεται η αντιμετώπιση συμβάντων βελτίωσης

από το λειτουργικό σύστημα),
- την πρόσθεση trackbacks στα παράθυρα και τη διαχείρηση τόσο απλών συμβάντων του ποντικιού όσο και εντολές πληκτρολογίου,
- την ανάγνωση και εγγαφή εικόνων από και προς κάποιο δίσκο ή μνήμη και
- την ανάγνωση βίντεο από κάμερα ή αρχείο και την εγγραφή βίντεο σε αρχείο.

– imgcodecs (Image file reading and writing, libopencv_imgcodecs300): Ανάγνωση και Εγγραφή Αρχείων Εικόνας. Αυτή η ενότητα της OpenCV καθιστά δυνατή την ανάγνωση και εγγραφή εικόνων από και προς το δίσκο ή τη μνήμη.

– imgproc (Image Processing, libopencv_imgproc300): Επεξεργασία Εικόνας. Αυτό το κομμάτι μπορεί να χρησιμοποιηθεί για φιλτράρισμα εικόνας, γεωμετρικούς και διάφορους άλλους μετασχηματισμούς εικόνας, συναρτήσεις σχεδιασμού, ColorMaps στην OpenCV, περιγραφή, ιστογράμματα, δομική ανάλυση και περιγραφείς σχήματος, ανάλυση κίνησης και παρακολούθηση αντικειμένων, εντοπισμό χαρακτηριστικών και εντοπισμό αντικειμένων.

– ml (Machine Learning, libopencv_ml300): Νοημοσύνη των Μηχανών. Η βιβλιοθήκη νοημοσύνης μηχανών (Machine Learning Library, MLL) είναι ένα σύνολο από κλάσεις και συναρτήσεις για στατιστική ταξινόμηση, οπισθοδρόμηση και ομαδοποίηση δεδομένων. Οι περισσότεροι αλγόριθμοι ταξινόμησης και οπισθοδρόμησης υλοποιούνται ως κλάσεις C++. Οι αλγόριθμοι έχουν διαφορετικά σύνολα χαρακτηριστικών (όπως την ικανότητα να χειριστούν χαμένες μετρήσεις ή κατηγορικές μεταβλητές εισόδου), επομένως υπάρχουν ελάχιστα κοινά σημεία μεταξύ των κλάσεων.Τα κοινά σημεία ορίζονται από την κλάση CvStatModel από την οποία παράγονται όλες οι κλάσεις ML. Το κομμάτι αυτό περιέχει στατιστικά μοντέλα, κανονικό ταξινομητή Bayes, K-πλησιέστερους γείτονες, μηχανές υποστήριξης διανυσμάτων, δέντρα αποφάσεων, ενίσχυση (boosting), ενισχυμένα δέντρα κλίσης, τυχαία δέντρα, μεγιστοποίηση προσδοκίας, νευρωνικά δίκτυα, δεδομένα ML.

– objdetect (Object Detection, libopencv_objdetect300): Ανίχνευση Αντικειμένων. Το κομμάτι αυτό περιέχει συναρτήσεις για ταξινόμηση καταρράκτη και λανθάνουσες μηχανές υποστήριξης διανυσμάτων.

– photo (Computational Photography, libopencv_photo300): Υπολογιστική Φωτογραφία. Περιέχει συναρτήσεις για inpainting (ανακατασκευή χαμένων ή φθαρμένων κομματιών εικόνων και βίντεο) και denoising (αφαίρεση "θορύβου" από την εικόνα).

– shape (Shape Distance and Matching, libopencv_shape300): Απόσταση και Αντιστοίχηση Σχημάτων. Η ενότητα περιέχει αλγορίθμους που ενσωματώνουν μια αντίληψη απόστασης σχήματος. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για αντιστοίχηση και ανάκτηση σχημάτων, ή για σύγκριση σχημάτων. Περιλαμβάνουν κοινές διεπαφές και διεπαφές απίστασης σχήματος, μετασχηματιστές και διεπαφές σχήματος, κοινή διεπαφή μήτρας κόστους για ιστογράμματα.

– stitching (Images stitching, libopencv_stitching300): Συρραφή Εικόνων. Περιλαμβάνει συναρτήσεις για το pipeline συρραφής, λειτουργικότητα υψηλού επιπέδου, την κάμερα, εύρεση χαρακτηριστικών και αντιστοίχηση εικόνων, εκτίμηση περιστροφής, αυτόματη βαθμονόμηση, διαστρέβλωση εικόνων, εκτίμηση γραμμής συναρμογής, αντιστάθμιση έκθεσης και αναδευτήρες εικόνων.

– superres (Super Resolution, libopencv_superres300): Μεγάλη Ευκρίνεια. Η ενότητα αυτή περιλαμβάνει ένα σύνολο συναρτήσεων και κλάσεων που μπορούν να χρησιμοποιηθούν για

την επίλυση του προβλήματος της ενίσχυσης της ευκρίνειας.

- video (Video analysis, libopencv_video300): Ανάλυση Βίντεο. Το κομμάτι αυτό περιέχει συναρτήσεις για ανάλυση κίνησης και παρακολούθηση αντικειμένων.
- videoio (Media I/O, libopencv_videoio300): Είσοδος/Έξοδος Πολυμέσων. Η ενότητα videoio παρέχει μια εύχρηστη διεπαφή για την ανάγνωση βίντεο από την κάμερα ή κάποιο αρχείο και την εγγραφή βίντεο σε αρχείο.
- videostab (Video stabilization, libopencv_videostab): Σταθεροποίηση Βίντεο. Η ενότητα σταθεροποίησης βίντεο περιέχει ένα σύνολο συναρτήσεων και κλάσεων που μπορούν να χρησιμοποιηθούν για την επίλυση του προβλήματος της σταθεροποίησης βίντεο. Περιλαμβάνει συναρτήσεις για εκτίμηση καθολικής κίνησης και για μεθόδους γρήγορου βήματος.
- ffmpeg (opencv_ffmpeg300): Περιλαμβάνει συναρτήσεις υποστήριξης FFMPEG για την OpenCV.

4.2 Αλγόριθμος και Πηγαίος Κώδικας της Εφαρμογής dFrame

Στην υποπαράγραφο αυτή παρουσιάζεται αναλυτικά ο πηγαίος κώδικας της εφαρμογής dFrame σε C++ χρήσει OpenCV και ο αλγόριθμος βάσει του οποίου υλοποιήθηκε. Ο κώδικας αναπτύχθηκε στο ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse IDE, με χρήση της εργαλειοθήκης για ανάπτυξη εφαρμογών στις γλώσσες C/C++ του Eclipse, CDT. Το μεγαλύτερο μέρος του κώδικα είναι ανεπτυγμένο στη γλώσσα C, με την προσθήκη κάποιων εντολών και συναρτήσεων από τη γλώσσα C++ (η οποία είναι συμβατή προς τα πάνω με τη C) καθώς και από τις βιβλιοθήκες όρασης υπολογιστών ανοιχτού κώδικα OpenCV.

4.2.1 Αλγόριθμος Εφαρμογής dFrame

Ο αλγόριθμος στον οποίο βασίζεται η υλοποιήση του προγράμματος dFrame υποθέτει ότι ο χρήστης δίνει ως παραμέτρους εισόδου στο πρόγραμμα κατά την εκτέλεσή του τα ονόματα ενός αρχείου βίντεο εισόδου, ενός αρχείου κειμένου εισόδου και ενός αρχείου κειμένου εξόδου, το οποίο θα δημιουργηθεί από την εφαρμογή. Όπως αναφέρθηκε και παραπάνω, το αρχείο βίντεο εισόδου πρέπει να περιλαμβάνει βίντεο από συνεδρία παιχνιδιού Super Mario Bros, στο οποίο έχουν καταγραφεί οι οπτικές αντιδράσεις και η συμπεριφορά του παίκτη. Το αρχείο κειμένου εισόδου με το όνομα που περνά ο παίκτης ως παράμετρο στο πρόγραμμα οφείλει να αντιστοιχεί στις μετρήσεις των χαρακτηριστικών του προσώπου (με τη μορφή κανονικοποιημένων συντεταγμένων 19 σημείων) για κάθε καρέ του βίντεο εισόδου. Πρόκειται για το αρχείο που προκύπτει από εκτέλεση της προηγούμενης εφαρμογής, FacialFeat, όταν αυτή δέχεται ως είσοδο το αρχείο βίντεο εισόδου που χρησιμοποιεί και η dFrame, ή όταν δημιουργήθηκε το αρχείο βίντεο αυτό ως έξοδος της FacialFeat μετά από εκτέλεση της λειτουργίας της για καταγραφή βίντεο από την κάμερα του υπολογιστή. Το αρχείο κειμένου εξόδου είναι ένα νέο αρχείο κειμένου που παράγεται από το πρόγραμμα αν αυτό επιλέξει ο χρήστης, με το όνομα που δόθηκε ως παράμετρος εισόδου. Περιέχει τις χρονικές στιγμές σε milliseconds και τις αντίστοιχες τιμές των χαρακτηριστικών του προσώπου του παίκτη για τα καρέ όπου παρατηρείται η μεγαλύτερη απόκλιση των συντεταγμένων των διαφόρων σημείων του προσώπου από τις τιμές αναφοράς που έχουν οριστεί μέσα στην εφαρμογή. Το αρχείο εξόδου αυτό αποθηκεύεται

στο μονοπάτι όπου βρίσκεται το .cpp αρχείο πηγαίου κώδικα αλλά και το εκτελέσιμο αρχείο της dFrame, στο οποίο πρέπει να είναι τοποθετημένα και δύο αρχεία εισόδου. Ο αλγόριθμος που ακολουθήθηκε για τον εντοπισμό των χαρακτηριστικών τιμών των σημείων του προσώπου, των χρονικών στιγμών και των καρέ στα οποία αυτές εμφανίζονται και την εξαγωγή τους στο αρχείο κειμένου εξόδου αλλά και σε αρχεία εικόνας, όπως θα δούμε παρακάτω, αποτελείται από τα εξής βήματα:

- Έλεγχος του αριθμού και των τύπων των παραμέτρων εισόδου του προγράμματος που δίνονται από το χρήστη.
- Ανάγνωση των αρχείων εισόδου (βίντεο και κειμένου) και αποθήκευσή τους μέσω μεταβλητών για περεταίρω ανάλυση.
- Ορισμός μονοδιάστατου πίνακα 44 στοιχείων που περιέχει τις τιμές αναφοράς που έχουμε επιλέξει για τον παίκτη του αρχείου βίντεο. Έχουμε 38 τιμές για τα 19 σημεία του προσώπου (δύο τιμές συντεταγμένων για κάθε σημείο) και τις έξι τιμές των καταστάσεων προσοχής.
- Προσπέλαση του αρχείου κειμένου εισόδου και καταμέτρηση των γραμμών του. Κάθε γραμμή του αρχείου κειμένου εισόδου αντιστοιχεί σε ένα καρέ του αρχείου βίντεο εισόδου και περιλαμβάνει τις τιμές των κανονικοποιημένων συντεταγμένων των σημείων για το αντίστοιχο καρέ. Στην αρχή κάθε γραμμής του αρχείου κειμένου αυτού βρίσκεται η τιμή των milliseconds του καρέ για τη χρονική στιγμή αυτή στο βίντεο. Συνεπώς, η καταμέτρηση των γραμμών του αρχείου γίνεται με σκοπό τον ορισμό του αριθμού των καρέ.
- Δημιουργία ενός μονοδιάστατου πίνακα για τα milliseconds. Το μέγεθος του πίνακα είναι ίσο με τον αριθμό των γραμμών που υπολογίστηκε στο προηγούμενο βήμα (δηλαδή τον αριθμό των καρέ) και οι τιμές του είναι οι τιμές των milliseconds για τη χρονική στιγμή κάθε καρέ.
- Δημιουργία ενός διδιάστατου πίνακα για τις τιμές των χαρακτηριστικών του προσώπου του παίκτη. Ο αριθμός των γραμμών του πίνακα είναι ίσος με τον αριθμό των γραμμών του αρχείου κειμένου (δηλαδή τον αριθμό των καρέ) και ο αριθμός των στηλών του πίνακα είναι ίσος με το πλήθος των χαρακτηριστικών του παίκτη (44, όπως στον πίνακα των τιμών αναφοράς).
- Προσπέλαση κάθε γραμμής του αρχείου κειμένου εισόδου και αποθήκευση των τιμών που περιέχει στους πίνακες που δημιουργήσαμε. Ο αλγόριθμος εξετάζει κάθε γραμμή του αρχείου κειμένου ξεχωριστά μέσω ενός βρόχου. Για κάθε εκτέλεση του βρόχου, αποθηκεύεται στο μονοδιάστατο πίνακα η πρώτη τιμή (τύπου double) της γραμμής που εξετάζεται, δηλαδή η τιμή των milliseconds για τη χρονική στιγμή του καρέ αυτού. Με τη χρήση ενός δείκτη για παράκαμψη της τιμής που αποθηκεύσαμε προχωράμε στην εξέταση της υπόλοιπης γραμμής, η οποία περιέχει τις τιμές των χαρακτηριστικών του προσώπου. Οι τιμές αυτές αποθηκεύονται στη γραμμή του διδιάστατου πίνακα η οποία αντιστοιχεί στη γραμμή του αρχείου κειμένου που εξετάζεται, και επομένως στο καρέ με τον ίδιο αριθμό. Στην τρέχουσα μορφή του προγράμματος και κυρίως για λόγους ελέγχου ορθής λειτουργίας του, μέσα στο βρόχο αυτό, μετά την αποθήκευση των τιμών των milliseconds και των χαρακτηριστικών του προσώπου στους πίνακες, οι τιμές αυτές εξάγονται και σε ένα νέο αρχείο κειμένου, το οποίο αν η εφαρμογή λειτουργήσει σωστά θα είναι ακριβώς το ίδιο με το αρχείο κειμένου εισόδου. Στη συνέχεια μηδενίζουμε και αυξάνουμε τους μετρητές και επαναλαμβάνουμε τη διαδικασία μέχρι να φτάσουμε στο τέλος του αρχείου κειμένου

εισόδου.

- Σύγκριση των τιμών των χαρακτηριστικών του προσώπου του παίκτη με τις τιμές αναφοράς και εντοπισμός των καρέ στα οποία παρουσιάζονται οι μεγαλύτερες αποκλίσεις. Πρόκειται για το κύριο αντικείμενο του αλγορίθμου αυτού και μας ενδιαφέρει συγκεκριμένα οποιαδήποτε απόκλιση από την αρχική κατάσταση αδρανείας του παίκτη –η οποία αντιπροσωπεύεται από τις τιμές αναφοράς που έχουμε επιλέξει για τις συντεταγμένες— μεγαλύτερη του 10%. Ο αλγόριθμος εξετάζει όλες τις τιμές των χαρακτηριστικών του προσώπου που έχουν αποθηκευθεί στον διδιάστατο πίνακα, σε σύγκριση με τις τιμές αναφοράς που βρίσκονται σε μονοδιάστατο πίνακα όπως ορίστηκαν νωρίτερα. Για αυτή τη διαδικασία χρησιμοποιείται ένας βρόχος όπου κάθε τιμή των συντεταγμένων των 19 σημείων συγκρίνεται με το 90% και το 110% κάθε αντίστοιχης τιμής αναφοράς. Αν η τιμή που εξετάζεται είναι μικρότερη του 90% ή μεγαλύτερη του 110% της τιμής αναφοράς, η τιμή των milliseconds που αντιστοιχεί στην γραμμή/καρέ αυτή αποθηκεύεται σε νέο μονοδιάστατο πίνακα, ο οποίος στο τέλος του βρόχου θα περιέχει όλες τις χρονικές στιγμές στις οποίες κάποιο χαρακτηριστικό του προσώπου εμφανίζει μεγαλύτερη του 10% απόκλιση. Κατά τον εντοπισμό των καρέ που μας ενδιαφέρουν, οι τιμές των milliseconds εγγράφονται και στο νέο αρχείο εξόδου, του οποίου το όνομα έδωσε ο χρήστης ως παράμετρο εισόδου στο πρόγραμμα. Η διαδικασία επαναλαμβάνεται για κάθε τιμή όλων των γραμμών/καρέ και τελειώνει όταν εξεταστεί και το τελευταίο στοιχείο της τελευταίας γραμμής του διδιάστατου πίνακα.
- Δημιουργία νέου διδιάστατου πίνακα, ο οποίος θα περιέχει τις τιμές των χαρακτηριστικών προσώπου του παίκτη για τα καρέ όπου παρατηρήθηκαν οι μεγαλύτερες του 10% αποκλίσεις. Ο αριθμός των γραμμών του πίνακα είναι ίσος με τον αριθμό των στοιχείων του μονοδιάστατου πίνακα που περιέχει τα milliseconds των χρονικών στιγμών των χαρακτηριστικών καρέ. Ο αριθμός των στηλών του πίνακα είναι και πάλι ίσος με το πλήθος των τιμών των χαρακτηριστικών προσώπου κάθε καρε, δηλαδή 44. Ο πίνακας δημιουργείται με χρήση ενός βρόχου, αποθηκεύοντας στις θέσεις του εκείνα τα στοιχεία του προηγούμενου διδιάστατου πίνακα τα οποία επιλέγουμε βάσει των milliseconds στα οποία αντιστοιχούν.
- Δημιουργία αρχείων εικόνων για τα καρέ μεγάλης απόκλισης και απεικόνιση των αντίστοιχων σημείων του προσώπου πάνω σε αυτά. Για το σκοπό αυτό κατασκευάζουμε ένα βρόχο που εκτελείται αριθμό φορών ίσο με το πλήθος των χρονικών στιγμών μεγάλης απόκλισης που έχουμε αποθηκεύσει. Ξεκινώντας από την πρώτη εμφάνιση μεγαλύτερης του 10% απόκλισης, πηγαίνουμε στο σημείο του αρχείου βίντεο εισόδου που μας υποδεικνύει η αντίστοιχη τιμή milliseconds και εξάγουμε το καρέ σε μια μεταβλητή. Μετά την αποθήκευσή του, ελέγχουμε το κατά πόσο υπάρχει τέτοιο καρέ (δηλαδή αν η χρονική στιγμή που μας δίνουν τα milliseconds υπάρχει μέσα στο αρχείο βίντεο εισόδου ή οχι). Στη συνέχεια, σε εμφωλευμένο βρόχο, διαβάζουμε όλα τα στοιχεία της γραμμής (καρέ) του διδιάστατου πίνακα των αποκλίσεων στην οποία βρισκόμαστε. Για κάθε δυάδα συντεταγμένων απεικονίζουμε ως μπλε δίσκους μικρών διαστάσεων στο καρέ το αντίστοιχο σημείο στο πρόσωπο, μέχρι το τέλος των στοιχείων της γραμμής. Η διαδικασία επαναλαμβάνεται για κάθε τιμή του δεύτερου μονοδιάστατου πίνακα των milliseconds (που ορίζει τις στιγμές των αποκλίσεων).

Σε αυτό το σημείο ο αλγόριθμος έχει πετύχει το βασικό του σκοπό, ελευθερώνει τις μεταβλητές που χρησιμοποιήθηκαν και τελειώνει. Η σωστή εκτέλεση του αλγορίθμου απαιτεί σωστές

παραμέτρους εισόδου από το χρήστη (ένα αρχείο κειμένου εισόδου, ένα αρχείο βίντεο εισόδου, ένα όνομα για νέο αρχείο κειμένου εξόδου). Παράγει ένα ή δύο αρχεία κειμένου εξόδου, ανάλογα με το αν ο χρήστης θέλει να ελέγξει τη σωστή λειτουργία του προγράμματος, και ένα σύνολο αρχείων εικόνας, τα οποία απεικονίζουν τα καρέ στα οποία τα χαρακτηριστικά προσώπου του παίκτη που εμφανίζουν μεγαλύτερη του 10% απόκλιση από τα σημεία αναφοράς. Τα σημεία αυτά σημειώνονται πάνω στο πρόσωπο του παίκτη σε κάθε αρχείο εικόνας.

Αυτό όπως είδαμε επιτυγχάνεται με απλές διαδικασίες, ξεκινώντας απο την ανάγνωση και προσπέλαση των αρχείων εισόδου, αποθηκεύουμε τα δεδομένα του αρχείου κειμένου εισόδου σε δύο πίνακες και συγκρίνουμε τις όλες τις τιμές των συντεταγμένων με τον πίνακα σημείων αναφοράς. Στη συνέχεια κρατάμε σε νέους πίνακες τις χρονικές στιγμές και τις τιμές συντεταγμένων που τηρούν τις προϋποθέσεις και τις χρησιμοποιούμε για τον εντοπισμό των χαρακτηριστικών καρέ στο αρχείο εισόδου βίντεο, την εξαγωγή τους ως νέα αρχεία εικόνας και την απεικόνιση των σημείων του προσώπου σε κάθε εικόνα.

4.2.2 Υλοποίηση του Αλγορίθμου και Πηγαίος Κώδικας της Εφαρμογής dFrame

Στην υποπαράγραφο αυτή παρουσιάζεται ο πηγαίος κώδικας της εφαρμογής dFrame, αναλύεται ο τρόπος που υλοποίεται ο αλγόριθμος και εξετάζονται οι εντολές και συναρτήσεις των γλωσσών προγραμματισμού C και C++ και των βιβλιοθηκών όρασης υπολογιστών ανοιχτού κώδικα OpenCV που χρησιμοποιήθηκαν σε αυτή τη διαδικασία. Παρουσιάζεται συνοπτικά και ο σωστός τρόπος εκτέλεσης του προγράμματος στο Eclipse.

Η εφαρμογή υλοποιήθηκε στην πλατφόρμα Eclipse IDE με χρήση της εργαλειοθήκης CDT για ανάπτυξη εφαρμογών σε C/C++, δημιουργώντας μια νέα εργασία (project) σε C/C++ με χρήση του συστήματος μεταγλώττισης MinGW (Minimalist GNU for Windows). Μέσω της διαδικασίας σύνδεσης του ολοκληρωμένου περιβάλλοντος ανάπτυξης Eclipse με το σύστημα MinGW και τις βιβλιοθήκες της OpenCV, με τον τρόπο που αναφέρθηκε σε προηγούμενη παράγραφο, εξασφαλίζουμε τα βασικά includes με τις βιβλιοθήκες και αρχεία headers(κεφαλιδών) που χρειαζόμαστε για την ανάπτυξη ενός οποιουδήποτε προγράμματος σε C ή C++ αλλά και πιο συγκεκριμένα τα προαπαιτούμενα για τις συναρτήσεις που θα χρησιμοποιήσουμε. Πιο συγκεκριμένα, χρησιμοποιούνται οι εξής βιβλιοθήκες των C/C++:

- stdio.h: Η κεφαλίδα stdio.h ορίζει τρεις τύπους μεταβλητών, διάφορες μακροεντολές και ποικίλες συναρτήσεις για την εκτέλεση εισόδου και εξόδου. Από αυτούς τους τρεις τύπους μεταβλητών, χρησιμοποιήθηκαν στον πηγαίο κώδικα της εφαρμογής dFrame οι size_t (αναπόσπαστος τύπος χωρίς πρόσημο που αφορά μέγεθος στη μνήμη και είναι αποτέλεσμα της λέξης κλειδί sizeof) και FILE (τύπος αντικειμένου κατάλληλος για αποθήκευση πληροφοριών για ροή αρχείου).
- stdlib.h: Η κεφαλίδα stdlib.h ορίζει τέσσερεις τύπους μεταβλητών (συμπεριλαμβανομένου του size_t), και διάφορες μακροεντολές και συναρτήσεις γενικής χρήσης, όπως τα malloc και free για δέσμευση και αποδέσμευση μνήμης.
- string.h: Η κεφαλίδα string.h ορίζει έναν τύπο μεταβλητών (size_t), μια μακροεντολή (NULL, η τιμή μιας μηδενικής σταθεράς δείκτη) και διάφορες συναρτήσεις για χειρισμό πινάκων χαρακτήρων.

Χρησιμοποιούνται επίσης οι εξής βιβλιοθήκες όρασης υπολογιστών της OpenCV:

- core.hpp: Η κεφαλίδα core.hpp περιέχει τις βασικές δομές που χρησιμοποιεί η OpenCV, βασικές δομές και λειτουργίες της C, δυναμικές δομές, λειτουργίες σε πίνακες, συναρτήσεις σχεδιασμού και διάφορα άλλα εργαλεία τα οποία θα μας χρειαστούν αργότερα. Περιέχει επίσης το namespace (χώρος ονόματος) cv που είναι απαραίτητο για την εκτέλεση προγραμμάτων C++ με OpenCV.
- highgui.hpp: Η κεφαλίδα highgui.hpp περιέχει τις διεπαφές χρήστη που μας επιτρέπουν την επεξεργασία εικόνας και βίντεο. Πιο συγκεκριμένα, θα μας χρειαστούν οι συναρτήσεις για ανάγνωση και εγγραφή εικόνας από και προς τη μνήμη καθώς και οι συναρτήσεις για ανάγνωση αρχείου βίντεο από τη μνήμη.
- imgproc.hpp: Η κεφαλίδα imgproc.hpp περιέχει τις συναρτήσεις επεξεργασίας εικόνας, όπως τις συναρτήσεις σχεδιασμού και χρωματισμού που μας είναι απαραίτητες για την απεικόνιση των σημείων του προσώπου στα αρχεία εικόνας εξόδου, ειδικότερα με τη μορφή μπλε δίσκων μικρών διαστάσεων.

Στη συνέχεια αναλύονται τα πιο σημαντικά τμήματα του πηγαίου κώδικα της εφαρμογής dFrame και παρουσιάζονται συνοπτικά λίγες λεπτομέρειες για τις εντολές, τις κλάσεις και τις συναρτήσεις που χρησιμοποιήθηκαν.

Αρχικοποιήσεις μη-εγγενών Αντικειμένων και Έναρξη του Προγράμματος

Τα includes στον πηγαίο κώδικα έχουν τη μορφή:

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
```

όπου είναι εύκολο να παρατηρήσει κανείς ότι δώσαμε στο πρόγραμμα το μονοπάτι μνήμης (directory) όπου βρίσκεται καθεμία από τις κεφαλίδες που θέλουμε να χρησιμοποιήσουμε. Όλες οι κεφαλίδες βρίσκονται στο φάκελο opencv2.

Στη συνέχεια, ορίζουμε το namespace που χρησιμοποιείται:

```
using namespace cv;
```

Τα Namespaces παρέχουν μια μέθοδο για να αποφεύγονται συγκρούσεις ονομάτων σε μεγάλες εργασίες. Τα σύμβολα που δηλώνονται μέσα σε ένα namespace μπλοκ τοποθετούνται σε ένα ονομασμένο πεδίο που τα αποτρέπει από το να μπερδευτούν με σύμβολα άλλων πεδίων με ακριβώς το ίδιο όνομα. Γενικά, επιτρέπονται πολλαπλά namespace μπλοκ με το ίδιο όνομα. Όλες οι δηλώσεις μέσα σε αυτά τα μπλοκ γίνονται στο ονομασμένο πεδίο.

Η παραπάνω εντολή δηλώνει ότι χρησιμοποιούμε το cv namespace (using namespace), στο οποίο είναι τοποθετημένες όλες οι κλάσεις και συναρτήσεις της OpenCV. Αν τυχόν χρειαστεί να χρησιμοποιηθούν κλάσεις ή συναρτήσεις με εξωτερικά ονόματα που συγκρούονται με άλλες βιβλιοθήκες όπως η STL, ο χρήστης μπορεί να χρησιμοποιήσει σαφείς προσδιοριστές (cv::) για να επιλύσει τη σύγκρουση ονομάτων.

Ορίζεται η συνάρτηση main για προγράμματα σε γλώσσα C/C++:

```
int main(int argc, char *argv[])
```

{

Η συνάρτηση αυτή είναι καθολική για το πρόγραμμα και ορίζει την αρχή του. Καλείται μετά την αρχικοποίηση των μη-εγγενών αντικειμένων και στη συγκεκριμένη περίπτωση είναι τύπου int (ακεραίου). Οι παράμετροι argc και argv της main χρησιμοποιούνται για να δώσει ο χρήστης τις παραμέτρους εισόδου στο πρόγραμμα. Το πρόγραμμα λαμβάνει τον αριθμό των παραμέτρων στο argc (argument count) και το διάνυσμα των παραμέτρων στο argv (argument vector). Το όνομα του προγράμματος αποτελεί την πρώτη παράμετρο και το τελευταίο στοιχείο του argv θα είναι ένας μηδενικός δείκτης (null). Στην εφαρμογή dFrame, ο αριθμός των παραμέτρων θα παίρνει την τιμή τέσσερα και οι παράμετροι θα είναι το όνομα του προγράμματος (dFrame), το όνομα του αρχείου κειμένου εισόδου, το όνομα του αρχείου κειμένου εξόδου και το όνομα του αρχείου βίντεο εισόδου. Σε άλλα προγράμματα που δε χρειάζονται παραμέτρους εισόδου, είναι εξίσου έγκυρη η δήλωση "int main() {...}".

Δηλώσεις και Αρχικοποιήσεις Μεταβλητών

Στον πηγαίο κώδικα της dFrame, προτού ξεκινήσουν οι διαδικασίες επεξεργασίας και η χρήση των διαφόρων συναρτήσεων πρέπει να οριστούν οι μεταβλητές που χρειαζόμαστε. Αμέσως μετά την έναρξη του προγράμματος με τη main, ακολουθεί το κομμάτι των δηλώσεων, δηλαδή ο ορισμός κάποιων μεταβλητών ακεραίου τύπου (int) που θα χρησιμοποιηθούν για διαφόρους σκοπούς αργότερα και η αρχικοποίηση στο μηδέν ορισμένων από αυτές:

```
int i=0, lines=0, lns=0, dt=0, x, y;
int arr, tsz, j;
int ns, vcnt;
int n, n1, total_n=0;//w
```

Πιο συγκεκριμένα, οι lines, lns θα χρησιμοποιηθούν για λειτουργίες σχετικές με τις γραμμές του αρχείου κειμένου εισόδου, οι j,i,dt,vcnt,ns είναι διάφοροι μετρητές, στην arr θα αποθηκευθεί το μέγεθος του πίνακα τιμών και στην tsz το μέγεθος του πίνακα χρόνων και οι n, n1, total_n χρησιμοποιούνται στην προσπέλαση των γραμμών του αρχείου κειμένου.

Ορίζονται επίσης οι εξής μεταβλητές:

```
char ch, *string;
float **vals, **dval, w, refv[44];
FILE *fpt, *fp, *wfp, *yaf;
Mat frame;
```

Ο πίνακας χαρακτήρων string θα χρησιμοποιηθεί για την επεξεργασία των γραμμών του αρχείου κειμένου εισόδου, οι διδιάστατοι πίνακες vals και dval τύπου float για την αποθήκευση των τιμών των χαρακτηριστικών προσώπου που περιέχονται στο αρχείο κειμένου εισόδου και των τιμών όπου εμφανίζεται μεγαλύτερη του 10% απόκλιση από τα σημεία αναφοράς, η w ως προσωρινή μεταβλητή για την αποθήκευση των τιμών αυτών και ο πίνακας (μεγέθους 44) refv για τον ορισμό των τιμών αναφοράς. Για την αποθήκευση των τιμών των χαρακτηριστικών του προσώπου επιλέχθηκαν μεταβλητές τύπου float, καθώς οι τιμές αυτές εμφανίζονται στο αρχείο κειμένου εισόδου (το οποίο παράγεται ως αρχείο κειμένου εξόδου της εφαρμογής FacialFeat) με τη μορφή πραγματικών αριθμών πολλών δεκαδικών ψηφίων, προσφέροντας έτσι μεγαλύτερη ακρίβεια στον υπολογισμό των συντεταγμένων των σημείων του προσώπου.

Στη συνέχεια, με τον τύπο αρχείου FILE ορίζονται οι δείκτες για τα αρχεία κειμένου που θα

επεξεργαστούμε. Οι fpr και fp χρησιμοποιούνται για την επεξεργασία του αρχείου κειμένου εισόδου, ο wfp για τη δημιουργία του αρχείου κειμένου εξόδου που θα περιέχει τις τιμές των milliseconds για τα χαρακτηριστικά καρέ και ο yaf για το αρχείο κειμένου που χρησιμοποιείται για έλεγχο της σωστής λειτουργίας του προγράμματος και περιέχει όλες τις τιμές με τη μορφή που εμφανίζονται στο αρχείο κειμένου εισόδου. Ο τύπος αντικειμένου FILE ορίζει μια ροή και περιέχει τις απαραίτητες πληροφορίες για τον έλεγχό της, συμπεριλαμβανομένου ενός δείκτη στον buffer της, το δείκτη θέσης και όλες τις ενδείξεις κατάστασής της.

Τέλος, σε αυτό το κομμάτι των αρχικοποιήσεων και δηλώσεων μεταβλητών ορίζεται και η κλάση frame τύπου Mat για την εξαγωγή των αρχείων εικόνας από το αρχείο βίντεο εισόδου. Η κλάση Mat της OpenCV αντιπροσωπεύει έναν πυκνό μονοκάναλο ή πολυκάναλο πίνακα n-διαστάσεων. Εκτός από την αποθήκευση έγχρωμων εικόνων, μπορεί να χρησιμοποιηθεί και για την αποθήκευση ασπρόμαυρων εικόνων, διανύσματων και πινάκων με πραγματικές ή πολύπλοκες τιμές, όγκων voxel, διανυσματικών πεδίων, συνόλων σημείων (σύννεφα) σε κάποιο σύστημα συντεταγμένων, tensors και ιστογραμμάτων (αν όμως θέλουμε να αποθηκεύσουμε ιστογράμματα πολύ υψηλών διαστάσεων θα ήταν καλύτερη η χρήση της κλάσης SparseMat).

Έλεγχος Παραμέτρων Εισόδου και Άνοιγμα Αρχείων

Έχοντας δηλώσει και αρχικοποιήσει τις μεταβλητές που θα χρησιμοποιήσουμε, προχωράμε στο κομμάτι ελέγχου των παραμέτρων εισόδου.

Όπως αναφέραμε νωρίτερα ο χρήστης πρέπει, συμπεριλαμβανομένου του ονόματος του προγράμματος, να περάσει τέσσερις παραμέτρους (με την εξής σειρά: όνομα προγράμματος, όνομα αρχείου κειμένου εισόδου, όνομα αρχείου κειμένου εξόδου, όνομα αρχείου βίντεο εισόδου) στο πρόγραμμα:

```
if (argc!=4) {
        printf( "usage: %s input-filename output-filename\n", argv[0]);
       exit(2);
}
```

Με αυτή τη συνθήκη (if) εξασφαλίζουμε ότι αν ο αριθμός των παραμέτρων argc δεν είναι ίσος με τέσσερα, το πρόγραμμα θα σταματήσει την εκτέλεσή του, επιστρέφοντας την τιμή δύο στη main. Με τη χρήση του printf εμφανίζεται στο τερματικό και το μήνυμα `"usage: dFrame input-filename output-filename input-video filename"`, το οποίο περιγράφει στο χρήστη το σωστό τρόπο εκτέλεσης του προγράμματος και περάσματος των παραμέτρων. Θα πρέπει να πληκτρολογήσει στο τερματικό ακριβώς, με κενά μεταξύ των λέξεων, ως εξής: dFrame [το όνομα του αρχείου κειμένου εισόδου] [το όνομα του αρχείου κειμένου εξόδου] [το όνομα του αρχείου βίντεο εισόδου]. Παραδείγματος χάριν, dFrame 1_A.txt outfile.txt 1_A.avi, όπου τα 1_A.txt και 1_A.avi βρίσκονται στο ίδιο directory με το εκτελέσιμο αρχείο. Εναλλακτικά, χρησιμοποιώντας την πλατφόρμα Eclipse, απλά προσθέτουμε τις παραμέτρους (1_A.txt outfile.txt 1_A.avi) στη διαμόρφωση εκτέλεσης (run configuration) του προγράμματος, έχοντας τοποθετήσει τα 1_A.txt και 1_A.avi στο χώρο εργασίας (workspace) του project μας. Στην εντολή printf χρησιμοποιείται ο προσδιοριστής %s για παρεμβολή μεταβλητής τύπου string (πίνακα χαρακτήρων), έτσι ώστε παίρνοντας την πρώτη τιμή του διανύσματος παραμέτρων, argv[0] να τυπωθεί στο μήνυμα το όνομα του προγράμματος (dFrame) που πληκτρολόγησε ο χρήστης στην εκτέλεση.

Στη συνέχεια, το πρόγραμμα ανοίγει τα αρχεία κειμένου εισόδου και εξόδου με τη συνάρτηση fopen και ελέγχει αν η διαδικασία έγινε σωστά:

```
if ((fpt = fopen(argv[1],"r")) == NULL){
       printf("Open infile error\n");
       exit(1);
```

```
}

if ((wfp = fopen(argv[2],"w")) == NULL){
       printf("Open outfile error\n");
       exit(1);
}

if ((yaf = fopen("yafile.txt","w")) == NULL){
       printf("Open file error\n");
       exit(1);
}
```

Αναθέτουμε στους δείκτες ροής τύπου FILE fpt και wfp τις τιμές του διανύσματος των παραμέτρων argv[1] και argv[2], δηλαδή τα ονόματα των αρχείων κειμένου εισόδου και εξόδου, με τη συνάρτηση fopen. Η συνάρτηση αυτή δέχεται δύο παραμέτρους, ένα όνομα αρχείου και μια σταθερά τύπου string που ορίζει τη λειτουργία της. Η fopen ανοίγει το αρχείο του οποίου το όνομα καθορίζεται από την πρώτη παράμετρο και το συνδέει με μια ροή που μπορεί να αναγνωρισθεί σε μελλοντικές λειτουργίες από το δείκτη FILE που επιστρέφει. Η δεύτερη παράμετρος ορίζει τις λειτουργίες που επιτρέπονται στη ροή και πως αυτές πραγματοποιούνται. Ο δείκτης που επιστρέφεται μπορεί να διαχωριστεί από το αρχείο καλώντας την fclose ή την freopen. Όλα τα ανοιγμένα αρχεία κλείνουν αυτόματα στον κανονικό τερματισμό του προγράμματος.

Στη συγκεκριμένη περίπτωση, η δεύτερη παράμετρος της fopen παίρνει την τιμή “r” για το δείκτη fpt του αρχείου εισόδου προς ανάγνωση. Αυτό σημαίνει ότι η συνάρτηση θα ανοίξει ένα αρχείο για λειτουργίες εισόδου και ότι το αρχείο αυτό πρέπει να υπάρχει ήδη. Για τον wfp και τον yaf—ο οποίος ορίζει το αρχείο κειμένου που χρησιμοποιείται για έλεγχο με το όνομα yafile.txt—η δεύτερη παράμετρος της fopen παίρνει την τιμή “w”, γεγονός το οποίο σημαίνει ότι η συνάρτηση θα δημιουργήσει ένα νέο αρχείο για λειτουργίες εξόδου. Αν υπάρχει ήδη αρχείο με το όνομα που ορίζεται στην πρώτη παράμετρο, τα περιεχόμενά του διαγράφονται και το αρχείο αντιμετωπίζεται ως ένα νέο άδειο αρχείο.

Για κάθε εκτέλεση της fopen, ελέγχουμε με μια συνθήκη (if) αν ο δείκτης που επιστρέφει είναι μηδενικός (NULL). Αυτό σημαίνει ότι προέκυψε κάποιο σφάλμα κατά το άνοιγμα του αρχείου και αν ισχύει, η main τερματίζει τη λειτουργία της επιστρέφοντας την τιμή 1 και ένα μήνυμα στο χρήστη που προσδιορίζει το αρχείο για το οποίο η συνάρτηση παρουσίασε σφάλμα (infile-αρχείο εισόδου, outfile-αρχείο εξόδου, απλό file για το αρχείο ελέγχου).

Μετά το άνοιγμα και τον έλεγχο των αρχείων κειμένου, ανοίγουμε το αρχείο βίντεο:

```
VideoCapture cap(argv[3]); // open the video file for reading

if ( !cap.isOpened() )  // if not success, exit program
{
     printf("Cannot open the video file");
     exit(-1);
}
```

Η κλάση VideoCapture της OpenCV χρησιμοποιείται για καταγραφή βίντεο από αρχεία βίντεο, ακολουθίες εικόνων ή κάμερες. Η κλάση παρέχει C++ API για καταγραφή βίντεο από κάμερα ή για ανάγνωση αρχείων βίντεο και ακολουθιών εικόνων και βρίσκεται στην κεφαλίδα videoio.hpp. Η παράμετρος που χρησιμοποιεί μπορεί να είναι μια σταθερά τύπου string με το όνομα του αρχείου, όπως στη συγκεκριμένη περίπτωση όπου το string με το όνομα του αρχείου έχει περασθεί ως παράμετρος εισόδου στο πρόγραμμα και βρίσκεται στη θέση argv[3] του διανύσματος παραμέτρων εισόδου της main. Η κλάση μπορεί να δεχθεί σαν δεύτερη παράμετρο έναν ακέραιο για την προτίμηση

API ή, εναλλακτικά, ως μονάδική παράμετρο έναν ακέραιο που αντιπροσωπεύει τον αριθμό που έχει ανατεθεί σε όποιον εξοπλισμό καταγραφής βίντεο διαθέτει ο υπολογιστής (0 για την προεπιλεγμένη κάμερα).

Έχοντας αναθέσει το αρχείο βίντεο στην VideoCapture cap, ελέγχουμε με μια συνθήκη και χρήση της συνάρτησης isOpened() της κλάσης, αν το άνοιγμα του αρχείου έγινε χωρίς σφάλματα. Η συνάρτηση isOpened επιστρέφει true αν έχει αρχικοποιηθεί η καταγραφή βίντεο, δηλαδή αν πέτυχε η προηγούμενη κλήση του κατασκευαστή VideoCapture ή της VideoCapture::open. Αν προέκυψε κάποιο σφάλμα, το πρόγραμμα ενημερώνει με μήνυμα το χρήστη ότι δεν μπορεί να ανοίξει το αρχείο βίντεο και τερματίζει, επιστρέφοντας την τιμή -1.

Αρχικοποίηση Πίνακα Τιμών Αναφοράς

Αυτές είναι οι τιμές που επιλέχθηκαν (στην περίπτωση του παραδείγματος 2_A, και σε σχόλιο οι αντίστοιχες για το παράδειγμα 1_A) ως συντεταγμένες των σημείων αναφοράς στο πρόσωπο του παίκτη, οι οποίες και αποθηκεύθηκαν στο μονοδιάστατο πίνακα 44 στοιχείων refv:

```
//--initialize reference array (TIMES ANAFORAS 2_A)
refv[0]=0.000000; refv[1]=0.000000; refv[2]=0.000000; refv[3]=0.000000; refv[4]=1.000000;
refv[5]=0.000000;      refv[6]=203.000000;      refv[7]=349.000000;      refv[8]=201.000000;
refv[9]=327.000000;
refv[10]=202.000000;    refv[11]=412.000000;    refv[12]=202.000000;    refv[13]=389.000000;
refv[14]=202.000000;
refv[15]=335.000000;    refv[16]=202.000000;    refv[17]=403.000000;    refv[18]=204.000000;
refv[19]=339.000000;
refv[20]=197.000000;    refv[21]=339.000000;    refv[22]=207.000000;    refv[23]=401.000000;
refv[24]=197.000000;
refv[25]=401.000000;    refv[26]=281.000000;    refv[27]=335.000000;    refv[28]=282.000000;
refv[29]=396.000000;
refv[30]=291.000000;    refv[31]=366.000000;    refv[32]=273.000000;    refv[33]=368.000000;
refv[34]=192.000000;
refv[35]=351.000000;    refv[36]=184.000000;    refv[37]=327.000000;    refv[38]=188.000000;
refv[39]=412.000000;
refv[40]=192.000000; refv[41]=391.000000; refv[42]=253.000000; refv[43]=370.000000;
/*TIMES ANAFORAS 1_A
 * 195.006134 350.192017 193.504654 327.062012 192.242157 413.986755 192.048447 390.145416
 * 193.858002 336.118073 193.204849 401.112854 196.838120 337.127014 189.856491 337.127014
 * 198.229507 402.066071 188.153061 404.066071 270.000000 337.323090 272.000000 397.468353
  *    277.044983  366.488312  262.046753  368.709106  177.954483  349.075684  177.924179
327.033020
 *   176.075928 414.031494 185.008453 390.233673 244.014008 370.886749*/
```

Προετοιμασία,Αναθέσεις για το Βρόχο Προσπέλασης του Αρχείου Κειμένου Εισόδου

Μετράμε τις γραμμές του αρχείου κειμένου εισόδου για τον υπολογισμό των μεγεθών των πινάκων και του πλήθους των καρέ:

```
//--count file lines

while(!feof(fpt)) {
      ch = fgetc(fpt);
      if(ch == '\n') { lns++; }
}
fclose(fpt);
```

Για την καταμέτρηση των γραμμών χρησιμοποιούμε ένα βρόχο while, ο οποίος τερματίζει όταν διαβαστεί ο τελευταίος χαρακτήρας του αρχείου κειμένου εισόδου. Μέσα στο βρόχο, χρησιμοποιούμε τη συνάρτηση fgetc(FILE *stream) η οποία φέρνει τον επόμενο χαρακτήρα (χαρακτήρας χωρίς πρόσημο) της ροης stream και προχωρά το δείκτη θέσης της. Αν ο χαρακτήρας που θα φέρει η συνάρτηση είναι ο χαρακτήρας της νέας γραμμής, ο μετρητής γραμμών lns που είχε αρχικοποιηθεί στην αρχή του προγράμματος στο μηδέν αυξάνεται κατά ένα και η διαδικασία επαναλαμβάνεται για τον επόμενο χαρακτήρα. Όταν ο βρόχος τερματίσει, η συνάρτηση μας έχει φέρει όλους τους χαρακτήρες που περιέχονται στο αρχείο κειμένου και στη μεταβλητή lns βρίσκεται ο συνολικός αριθμός γραμμών του αρχείου αυτού. Στη συνέχεια μηδενίζουμε το δείκτη fpt, ο οποίος τώρα βρίσκεται στο τέλος του αρχείου, με χρήση της εντολής fclose(fpt).

Έχοντας διαθέσιμο τον αριθμό των γραμμών του αρχείου, μπορούμε να υπολογίσουμε τα μεγέθη των πινάκων που θα χρησιμοποιήσουμε, να δεσμέυσουμε τον αντίστοιχο χώρο στη μνήμη και να ορίσουμε κάποιες νέες βοηθητικές μεταβλητές:

```
tsz = lns;
```

Το μέγεθος του πίνακα χρόνου, tsz, παίρνει την τιμή της lns, γίνεται δηλαδή ίσο με τον αριθμό των γραμμών του αρχείου. Βάσει αυτού, ορίζονται τα μεγέθη των πινάκων χρόνου και δεσμεύεται ο αντίστοιχος χώρος στη μνήμη:

```
int *msecs = (int*)malloc(tsz*sizeof(int));
int *devtimes = (int*)malloc(tsz*sizeof(int)); //xronikes stigmes megalis apoklisis
int *dptr = (int*)malloc(tsz*sizeof(int)); //xronikes stigmes megalis apoklisis

for (ns=0; ns<tsz; ns++)
      msecs[ns]=0;
arr = 44;
```

Το μέγεθος των τριών μονοδιάστατων πινάκων ακεραίων που θα χρησιμοποιήσουμε για την αποθήκευση των τιμών χρόνου (milliseconds) τίθεται ίσο με τον αριθμό των γραμμών του αρχείου κειμένου που υπολογίσαμε. Ο πρώτος πίνακας msec θα χρησιμοποιήσει όλες αυτές τις θέσεις για αποθήκευση των milliseconds κάθε γραμμής/καρέ. Ο πίνακας devtimes το πιθανότερο είναι ότι θα έχει πραγματικό μέγεθος μικρότερο του tsz αλλά δεσμεύουμε τόση μνήμη όση και για τον msec. Στον πίνακα αυτόν θα αποθηκευθούν οι τιμές των milliseconds στα οποία οι τιμές των χαρακτηριστικών του προσώπου του παίκτη παρουσιάζουν μεγαλύτερη του 10% απόκλιση από τα σημεία αναφοράς όπως ορίστηκαν προηγουμένως. Το πλήθος των τιμών αυτών το πιθανότερο είναι ότι θα είναι αρκετά μικρότερο του πλήθους όλων των milliseconds, θεωρητικά όμως η μέγιστη τιμή που μπορεί να πάρει είναι η ίδια (tsz) και επομένως πρέπει να δεσμευτούν οι αντίστοιχες θέσεις μνήμης. Το ίδιο πρέπει να γίνει και για τον πίνακα dptr, ο οποίος θα χρησιμοποιηθεί για την αποθήκευση των θέσεων/μετρητών των τιμών που αναζητάμε στον πίνακα και θα έχει το ίδιο μέγεθος με τον πίνακα devtimes, με μέγιστη τιμή την τιμή του tsz.

Η δέσμευση των θέσεων μνήμης για τους πίνακες αυτούς γίνεται με χρήση της συνάρτησης της C για δέσμευση μπλοκ μνήμης (memory block allocation), void* malloc (size_t size). Η συνάρτηση αυτή δεσμεύει ένα μπλοκ από bytes πλήθους size και επιστρέφει ένα δείκτη στην αρχή αυτού του μπλοκ. Το περιεχόμενο του νέου δεσμευμένου μπλοκ δεν αρχικοποιείται, παραμένοντας με ασαφείς τιμές. Αν το size είναι ίσο με μηδέν, η τιμή επιστροφής εξαρτάται από την υλοποίηση της συγκεκριμένης βιβλιοθήκης (μπορεί να είναι ή να μην είναι ένας δείκτης NULL), αλλά ο δείκτης που επιστρέφεται δεν μπορεί να αναχθεί. Στην κλήση της malloc χρησιμοποιείται και η λειτουργία της C sizeof object/(type), η οποία επιστρέφει το μέγεθος του αντικειμένου ή του τύπου δεδομένων που της

δίνουμε σαν παράμετρο.

Στη συγκεκριμένη περίπτωση, το πλήθος των bytes που δεσμεύονται τελικά μέσω της malloc για κάθε πίνακα είναι το γινόμενο του μεγέθους του τύπου int των στοιχείων του πίνακα (sizeof(int)) και του tsz. Επειδή η malloc είναι τύπου void και οι πίνακές μας τύπου int, κάνουμε type casting με τη μέθοδο (int*) στην κλήση της malloc. Το type casting είναι ένας τρόπος μετατροπής ενός τύπου δεδομένων σε έναν άλλον τύπο δεδομένων.

Ακολουθεί η αρχικοποίηση των τιμών του πίνακα msec στο μηδέν με έναν απλό βρόχο for και η μεταβλητή arr, που δείχνει τον αριθμό των στηλών του διδιάστατου πίνακα που περιέχει τις τιμές των χαρακτηριστικών προσώπου, τίθεται ίση με 44 που είναι και ο αριθμός των στοιχείων κάθε γραμμής του αρχείου κειμένου μετά την τιμή των milliseconds.

Δεσμεύεται επίσης η μνήμη για τον πίνακα string και το διδιάστατο πίνακα τιμών χαρακτηριστικών προσώπου:

```
string = (char*) malloc (510);

if ((vals = (float **)malloc(sizeof(float *)*tsz))==NULL){
       printf("Out of memory - vals/rows\n");
       exit(-2);
}
for(i = 0; i < tsz; i++){
       if((vals[i] = (float*)malloc(sizeof(float)*arr))==NULL){
              printf("Out of memory - vals/columns\n");
              exit(-3);
       }
}
```

Για τον πίνακα χαρακτήρων δεσμεύονται μέσω της malloc (και type casting για τον τύπο δεδομένων char) 510 bytes έτσι ώστε να εξασφαλίσουμε ότι αργότερα θα χωρέσει ολόκληρη μια γραμμή του αρχείου κειμένου εισόδου. Η δέσμευση της μνήμης για τον διδιάστατο τύπου float πίνακα vals με tsz γραμμές και arr(44) στήλες, στον οποίο θα αποθηκευθούν οι τιμές των χαρακτηριστικών του προσώπου του παίκτη, γίνεται σε δύο στάδια (γραμμές και στήλες). Δεσμεύουμε με χρήση της malloc αριθμό bytes ίσο με το γινόμενο του tsz και του sizeof(float *) για πίνακα floats, κάνοντας και type casting για διδιάστατο πίνακα τύπου float (float **). Στη συνέχεια χρησιμοποιούμε ένα βρόχο for, tsz επαναλήψεων, στον οποίο καλούμε τη malloc για κάθε στοιχείο που δεσμεύσαμε στην προηγούμενη κλήση. Σε αυτήν την κλήση η malloc δέχεται type casting μονοδιάστατου πίνακα float (float*) και δεσμεύει για κάθε γραμμή 44 στήλες. Ο αριθμός byte για καθεμία είναι ίσος με το γινόμενο του arr και του sizeof(float). Με αυτόν τον τρόπο, δεσμεύτηκε μνήμη για το διδιάστατο πίνακα vals, αντιμετωπίζοντάς τον ως ένα μονοδιάστατο πίνακα float του οποίου τα στοιχεία είναι μονοδιάστατοι πίνακες float.

Για κάθε κλήση της malloc για δέσμευση μνήμης τόσο για τις γραμμές όσο και για τις στήλες του πίνακα vals ελέγχουμε, με μια συνθήκη (if), αν ο δείκτης που θα μας επιστρέψει η malloc είναι μηδενικός (NULL). Σε αυτήν την περίπτωση έχει προκύψει κάποιο σφάλμα μνήμης, η εφαρμογή τερματίζει και επιστρέφει την τιμή -2 αν η malloc απέτυχε για τις γραμμές ή την τιμή -3 για τις στήλες του πίνακα, καθώς και ένα αντίστοιχο μήνυμα στο χρήστη. Ο έλεγχος αυτός γίνεται και για ευκολότερο debugging, καθώς προσδιορίζει ποια διαδικασία σχετική με τη μνήμη απέτυχε και βοηθάει στην αποφυγή του πιο ασαφούς μηνύματος segmentation fault στο λειτουργικό σύστημα,για παραβίαση πρόσβασης στη μνήμη.

Σαν προετοιμασία για την εκτέλεση του βρόχου προσπέλασης του αρχείου κειμένου και εξαγωγής των δεδομένων του, ανοίγουμε ξανά με την fopen για ανάγνωση το αρχείο κειμένου εισόδου με το δείκτη fp και αρχικοποιούμε τους μετρητές στο μηδέν:

```
fp = fopen(argv[1],"r");
vcnt=i=0;
```

Κύριος Βρόχος Προσπέλασης Αρχείου Εισόδου και Εξαγωγής των Δεδομένων

Ο βρόχος που ακολουθεί χρησιμοποιείται για την προσπέλαση του αρχείου κειμένου εισόδου και την εξαγωγή των δεδομένων του, εκ των οποίων οι τιμές των milliseconds αποθηκεύονται στο μονοδιάστατο πίνακα msecs και οι τιμές των χαρακτηριστικών του προσώπου παίκτη στο διδιάστατο πίνακα vals:

```
while((fgets(string,510,fp))!=NULL){
      sscanf(string, "%d %n", &msecs[lines], &n1);
      fprintf(yaf,"%d ", msecs[lines]);
      total_n = n1;
      while((1 == sscanf(string + total_n, "%f %n", &w, &n)) && (i<44)) {
            total_n += n;
            vals[lines][i] = w;
            fprintf(yaf, "%f ", vals[lines][i]);
            i++;
      } fprintf(yaf,"\n");
      i = 0; total_n = 0;
      lines++;
}
```

Ο τερματισμός του βρόχου while εξαρτάται από την τιμή της συνάρτησης char * fgets (char * str, int num, FILE * stream) της C (get string from stream). Η συνάρτηση αυτή διαβάζει χαρακτήρες από τη ροή αρχείου stream και τους αποθηκεύει ως C string στον πίνακα str μέχρι να συμβεί όποιο από τα εξής συμβεί πρώτο: να διαβαστούν (num-1) χαρακτήρες, να φτάσει σε χαρακτήρα νέας γραμμής ή να φτάσει στο τέλος του αρχείου (end-of-file). Ένας χαρακτήρας νέας γραμμής (newline) θα αναγκάσει την fgets να σταματήσει την ανάγνωση από το αρχείο, αλλά θεωρείται έγκυρος χαρακτήρας από τη συνάρτηση και περιλαμβάνεται στο string που αντιγράφεται στον πίνακα str. Ένας μηδενικός χαρακτήρας (null) τερματισμού επισυνάπτεται αυτόματα μετά την αντιγραφή των χαρακτήρων στο str. Η συνάρτηση fgets είναι αρκετά διαφορετική από τη συνάρτηση gets. Σε αντίθεση με την gets, η fgets όχι μόνο δέχεται μια ροή (stream) ως παράμετρο, αλλά επιτρέπει επίσης τον ορισμό του μεγίστου μεγέθους του πίνακα str και περιλαμβάνει οποιοδήποτε χαρακτήρα νέας γραμμής στο string. Στην παράμετρο stream μπορεί να δοθεί και η τιμή stdin έτσι ώστε η συνάρτηση να διαβάσει από την κανονική είσοδο (standard input).

Στο βρόχο while, η fgets χρησιμοποιείται για να διαβάσει μία-μία τις γραμμές του αρχείου κειμένου εισόδου και να αποθηκεύει την τρέχουσα στον πίνακα χαρακτήρων string. Όταν η συνάρτηση μας φέρει την τελευταία γραμμή του αρχείου και στη συνέχεια πάρει την τιμή NULL, ο βρόχος τερματίζεται.

Για την επεξεργασία του πίνακα string χρησιμοποιείται η συνάρτηση int sscanf (const char * s, const char * format,....) της C (read formatted data from string). Διαβάζει δεδομένα από τον πίνακα χαρακτήρων s και τα αποθηκεύει ανάλογα με την παράμετρο format στις τοποθεσίες που δίνονται από τις πρόσθετες παραμέτρους. Χρησιμοποιείται όπως η scanf αλλά διαβάζει από τον πίνακα s αντί της κανονικής εισόδου (stdin). Οι πρόσθετες παράμετροι πρέπει να δείχνουν σε ήδη τοποθετημένα στη μνήμη αντικείμενα του τύπου που ορίζεται από τον αντίστοιχο προσδιοριστή μορφής στο format string. Οι παράμετροι που χρησιμοποιεί η συνάρτηση είναι:

- s: string C το οποίο επεξεργάζεται η συνάρτηση ως πηγή ανάκτησης δεδομένων,
- format: string C το οποίο περιέχει ένα string μορφής που χρησιμοποιεί τους ίδιους

προσδιορισμούς όπως το format της scanf. Σε αυτό το κομμάτι της εφαρμογής μας ενδιαφέρουν οι προσδιοριστές %f (για εξαγωγή χαρακτήρων τύπου float-οι τιμές των χαρακτηριστικών προσώπου), %d (για εξαγωγή δεκαδικών ακεραίων-οι τιμές χρόνου σε milliseconds) και %n (για την αποθήκευση του πλήθους των χαρακτήρων που έχουν διαβαστεί ως τώρα από το s).

- ...(πρόσθετες παράμετροι): με βάση το string format, η συνάρτηση μπορεί να περιμένει μια ακολουθία πρόσθετων παραμέτρων, καθεμία από τις οποίες θα περιέχει ένα δείκτη σε δεσμευμένη μνήμη όπου αποθηκεύεται η μορφή των χαρακτήρων που εξήχθησαν με την κατάλληλη μορφή. Πρέπει να υπάρχουν τουλάχιστον τόσες τέτοιες παράμετροι όσες οι τιμές που αποθηκεύονται από τους προσδιοριστές format. Επιπλέον παράμετροι αγνοούνται από τη συνάρτηση.

Αν η sscanf επιτύχει, επιστρέφει τον αριθμό των αντικειμένων στη λίστα παραμέτρων που γέμισε επιτυχώς. Αυτός ο αριθμός μπορεί να είναι ίσος με το αναμενόμενο πλήθος αντικειμένων ή, σε περίπτωση αποτυχίας αντιστοίχισης, μικρότερος από το αναμενόμενο ή ακόμα και μηδέν. Σε περίπτωση σφάλματος εισόδου πριν την εξέταση δεδομένων, επιστρέφεται EOF (τέλος αρχείου, end-of-file).

Η πρώτη χρήση της συνάρτησης sscanf στο βρόχο δέχεται τις παραμέτρους μορφής %d και %n. Με την παράμετρο %d η συνάρτηση εξάγει τον πρώτο ακέραιο δεκαδικό που θα βρει στον πίνακα string, στον οποίο βρίσκεται η τρέχουσα γραμμή του αρχείου κειμένου. Ο αριθμός αυτός είναι τα milliseconds του καρέ που εξετάζεται μέσω της γραμμής και αποθηκεύεται στην τρέχουσα θέση του πίνακα msecs, την οποία υποδεικνύει ο μετρητής lines (&msecs[lines]). Ο προσδιοριστής μορφής %n χρησιμοποιείται για την αποθήκευση, στη θέση &n1, του πλήθους των χαρακτήρων του πίνακα string που διαβάστηκαν μέχρι τώρα.

Στη συνέχεια καλούμε τη συνάρτηση fprintf για εξαγωγή των τιμών milliseconds που αποθηκεύονται μέσω του δείκτη yaf στο αρχείο κειμένου yafile.txt που χρησιμοποιείται για έλεγχο της σωστής εξαγωγής δεδομένων από το αρχείο κειμένου εισόδου. Η συνάρτηση της C, int fprintf (FILE * stream, const char * format, …) εγγράφει το string C που υποδεικνύει ο πίνακας format στη ροή stream. Αν ο πίνακας format περιέχει προσδιοριστές μορφής (υποακολουθίες που ξεκινούν με %), οι επιπλέον παράμετροι που ακολουθούν τον format διαμορφώνονται και εισάγονται στο string που προκύπτει, αντικαταστώντας τους αντίστοιχους προσδιοριστές. Μετά την παράμετρο format, η συνάρτηση περιμένει τουλάχιστον τόσες επιπλέον παραμέτρους όσες ορίστηκαν στο format. Αν η συνάρτηση επιτύχει, επιστρέφεται ο συνολικός αριθμός χαρακτήρων που εγγράφονται. Αν συμβεί σφάλμα εγγραφής, ορίζεται ο δείκτης λάθους (ferror) και επιστρέφεται ένας αρνητικός αριθμός. Αν διαβάζοντας ευρείς χαρακτήρες συμβεί σφάλμα κωδικοποίησης χαρακτήρα πολλαπλών bytes, η errno παιρνει την τιμή EILSEQ και επιστρέφεται ένας αρνητικός αριθμός.

Γίνεται επίσης ανάθεση της τιμής του πλήθους χαρακτήρων που βρίσκεται n1, που διαβάστηκαν για την εξαγωγή των milliseconds, στη μεταβλητή total_n, η οποία θα διευκολύνει την ανάγνωση και εξαγωγή των τιμών των χαρακτηριστικών του προσώπου του παίκτη στον επόμενο, εμφωλευμένο, βρόχο while.

Για την εξαγωγή των χαρακτηριστικών του προσώπου του παίκτη στο διδιάστατο πίνακα που έχουμε ορίσει, χρησιμοποιείται ένας εμφωλευμένος βρόχος while, του οποίου η συνθήκη τερματισμού εξαρτάται από την επεξεργασία του πίνακα string μέσω της συνάρτησης sscanf καθώς και από τον μετρητή (i) που περιορίζει τον αριθμό επαναλήψεων του βρόχου σε το πολύ 44, όσα είναι και τα στοιχεία κάθε γραμμής του διδιάστατου πίνακα των χαρακτηριστικών προσώπου παίκτη. Ο βρόχος δηλαδή, εφόσον ο μετρητής αυτός έχει τιμή μικρότερη του 44, συνεχίζει να εκτελείται όσο η κλήση της sscanf για την επεξεργασία του string επιστρέφει την τιμή ένα.

Η πρώτη παράμετρος της sscanf τίθεται ίση με string + total_n, με σκοπό την αποφυγή της επανεξέτασης των πρώτων [total_n] χαρακτήρων του πίνακα string. Ο δείκτης αυτός αναγκάζει τη συνάρτηση να ξεκινήσει τη λειτουργία της στο σημείο string + total_n του πίνακα και επομένως να προσπεράσει τους χαρακτήρες που αντιστοιχούν στην τιμή των milliseconds της γραμμής που εξετάζεται και να περάσει στην εξαγωγή των χαρακτήρων που αποτελούν τις τιμές χαρακτηριστικών προσώπου. Για την κλήση της συνάρτησης χρησιμοποιήθηκαν οι προσδιοριστές %f και %n. Με αυτόν τον τρόπο αποθηκεύεται στην προσωρινή μεταβλητή w η τρέχουσα τιμή της γραμμής και στη μεταβλητή n ο μετρητής χαρακτήρων που θα χρειαστεί για να διαβάσουμε την επόμενη τιμή της γραμμής.

Στη συνέχεια η τιμή της μεταβλητής total_n αυξάνεται κατά n έτσι ώστε ο δείκτης στο string που θα χρησιμοποιήσει η sscanf να δείχνει στον επόμενο χαρακτήρα της γραμμής του αρχείου κειμένου. Η τιμή της τρέχουσας θέσης του διδιάστατου πίνακα χαρακτηριστικών προσώπου vals (η οποία προσδιορίζεται από τους μετρητές lines και i) τίθεται ίση με τον αριθμό float w που επέστρεψε η sscanf (vals[lines][i]=w). Χρησιμοποιείται επίσης η συνάρτηση fprintf για την εξαγωγή στο αρχείο κειμένου yafile.txt κάθε τιμής που αποθηκεύεται στον πίνακα και ο μετρητής i αυξάνεται κατά ένα, έτσι ώστε η εκτέλεση του βρόχου while να σταματήσει όταν διαβάσουμε 44 τιμές από τον πίνακα string.

Έχοντας πλέον διαβάσει μια γραμμή του αρχείου κειμένου και αποθηκεύσει τις τιμές των χαρακτηριστικών προσώπου που περιέχει, ο εμφωλευμένος βρόχος while κλείνει, μηδενίζουμε τους μετρητές total_n και i, αυξάνουμε το μετρητή γραμμών lines κατά ένα και ο μητρικός βρόχος επανεκτελείται για την αποθήκευση της επόμενης γραμμής του αρχείου κειμένου στον πίνακα string. Επομένως, για κάθε εκτέλεση του αρχικού βρόχου, εισάγουμε μία τιμή milliseconds στο μονοδιάστατο πίνακα msecs και 44 τιμές χαρακτηριστικών προσώπου σε μια γραμμή (ή μια τιμή σε κάθε στήλη) στο διδιάστατο πίνακα vals. Όπως είδαμε, όταν η fgets επιστρέψει την τιμή null ο αρχικός βρόχος while τερματίζει καθώς, αν δεν προέκυψε κάποιο σφάλμα, φτάνουμε στο τέλος της ροής του αρχείου κειμένου.

Αναζήτηση και Αποθήκευση των Χρονικών Στιγμών Μεγάλης Απόκλισης

Έχοντας αποθηκεύσει όλα τα δεδομένα του αρχείου κειμένου εισόδου στους πίνακες msecs για τις χρονικές στιγμές και vals για τα χαρακτηριστικά του προσώπου, προχωράμε στη διαδικασία εντοπισμού των καρέ που μας ενδιαφέρουν, μέσω των χρονικών στιγμών (milliseconds) όπου κάποιο από τα χαρακτηριστικά προσώπου εμφανίζει μεγαλύτερη του 10% απόκλιση από τις επιλεγμένες τιμές αναφοράς:

```
for(i=0; i<lines; i++){
      for(j=0; j<44; j++){
            if ((vals[i][j]>refv[j]*1.1)||(vals[i][j]<refv[j]*0.9)){
                  vcnt++;
                  if (vcnt == 44){
                        devtimes[dt] = msecs[i]; dptr[dt] = i; dt++;
                        fprintf(wfp, "millisecs[%d] = %d\n", i, msecs[i]);
                        vcnt = 0;
                  }
            }
      }
}
```

Ο βρόχος αυτός (for) χρησιμοποιείται για την εύρεση των χαρακτηριστικών καρέ που μας απασχολούν, που είναι και ο βασικός σκοπός της εργασίας και επομένως της εφαρμογής dFrame. Στα καρέ αυτά, όλες οι τιμές των χαρακτηριστικών του προσώπου, με τη μορφή κανονικοποιημένων συντεταγμένων σημείων, παρουσιάζουν μεγαλύτερη του 10 % απόκλιση από τις συντεταγμένες των επιλεγμένων σημείων αναφοράς. Τα σημεία αναφοράς επιλέχθηκαν έτσι ώστε καθορίζουν την αρχική στάση του κεφαλιού του παίκτη και την ουδέτερη ή αδρανή κατάσταση του προσώπου του. Για το σκοπό αυτό θα χρειαστούμε τους αντίστοιχους πίνακες, δηλαδή το διδιάστατο πίνακα vals με τα δεδομένα για τα χαρακτηριστικά του προσώπου που διαβάστηκαν και εξήχθησαν από το αρχείο κειμένου εισόδου και το μονοδιάστατο πίνακα refv που περιέχει τις τιμές των σημείων αναφοράς που παρατέθηκαν προηγουμένως.

Ο εξωτερικός βρόχος for εκτελείται αριθμό φορών ίσο με την τιμή της μεταβλητής lines (η οποία περιέχει τον αριθμό των γραμμών του πίνακα vals) έτσι ώστε να εξεταστούν όλες οι γραμμές του διδιάστατου πίνακα vals. Για κάθε γραμμή του πίνακα, δηλαδή κάθε εκτέλεση του εξωτερικού βρόχου, εκτελείται ο εμφωλευμένος βρόχος for 44 φορές, μια φορά για κάθε στοιχείο της γραμμής του πίνακα vals ή, ισοδύναμα, μια φορά για κάθε στήλη του πίνακα vals. Για κάθε στοιχείο της τρέχουσας γραμμής του πίνακα vals εκτελείται στον εμφωλευμένο βρόχο μια συνθήκη (if). Η συνθήκη αυτή ελέγχει αν το στοιχείο του vals που εξετάζεται διαφέρει κατά ποσοστό μεγαλύτερο του 10% από την αντίστοιχη τιμή αναφοράς του πίνακα refv, χρησιμοποιώντας τον λογικό τελεστή "||", ο οποίος πραγματοποιεί τη λογική πράξη "or" (ή) μεταξύ δύο ανισώσεων. Αν οποιαδήποτε από τις δύο ανισώσεις επιστρέψει τη λογική τιμή "true" ή 1, θα μας επιστρέψει την τιμή true και ο λογικός τελεστής or. Αν και οι δύο ανισώσεις επιστρέψουν τη λογική τιμή "false" ή 0, τότε και ο τελεστής or θα επιστρέψει την τιμή false. Η λογική συνθήκη στη συγκεκριμένη περίπτωση εκτελείται με τον εξής τρόπο:

- εκτελείται η ανίσωση (vals[i][j] > refv[j]*1.1). Αν η τιμή του τρέχοντος στοιχείου του πίνακα vals (vals[i][j]) είναι μεγαλύτερη του 110 % της τιμής του αντίστοιχου σημείου αναφοράς (refv[j]), η ανίσωση επιστρέφει την τιμή true (αληθής). Αυτό σημαίνει ότι το στοιχείο που εξετάζουμε είναι μεγαλύτερο κατά ποσοστό μεγαλύτερο του 10% από το αντίστοιχο στοιχείο του πίνακα τιμών αναφοράς. Ο λογικός τελεστής "||" επιστρέφει την τιμή true, επομένως ικανοποιείται η συνθήκη του if και προχωράμε στην εκτέλεση των εντολών που περιέχει. Αν η τιμή του τρέχοντος στοιχείου είναι μικρότερη του 110% της αντίστοιχης τιμής αναφοράς, η ανίσωση επιστρέφει την τιμή false και προχωράμε στο επόμενο βήμα,
- εκτελείται η ανίσωση (vals[i][j] < refv*0.9). Αν η τιμή του τρέχοντος στοιχείου του πίνακα vals (vals[i][j]) είναι μικρότερη του 90 % της τιμής του αντίστοιχου σημείου αναφοράς (refv[j]), η ανίσωση επιστρέφει την τιμή true (αληθής). Αυτό σημαίνει ότι το στοιχείο που εξετάζουμε είναι μικρότερο κατά ποσοστό μεγαλύτερο του 10% από το αντίστοιχο στοιχείο του πίνακα τιμών αναφοράς. Ο λογικός τελεστής "||" επιστρέφει την τιμή true, επομένως ικανοποιείται η συνθήκη του if και προχωράμε στην εκτέλεση των εντολών που περιέχει. Αν η τιμή του τρέχοντος στοιχείου είναι μεγαλύερη του 90% αλλά και μικρότερη του 110% της αντίστοιχης τιμής αναφοράς, η ανίσωση επιστρέφει την τιμή false και προχωράμε στο επόμενο βήμα.
- Αν και οι δύο ανισώσεις που εφαρμόσουμε επιστρέψουν τιμές false (ψευδής), σημαίνει ότι το στοιχείο του πίνακα vals που εξετάζουμε έχει τιμή ανάμεσα στο 90% και το 110% του αντίστοιχου στοιχείου του πίνακα τιμών αναφοράς και δε μας ενδιαφέρει. Η συνθήκη του if δεν ικανοποιείται, καθώς και ο τελεστής or θα μας επιστρέψει την τιμή false και το

πρόγραμμα προχωράει στην επόμενη εκτέλεση του εμφωλευμένου βρόχου, για την εξέταση του επόμενου στοιχείου του vals.

Αν οι συνθήκες ικανοποιούνται, προχωράμε στην αποθήκευση των χρονικών στιγμών που αντιστοιχούν στα σημεία που παρατηρήθηκαν οι αποκλίσεις που μας ενδιαφέρουν. Αυτό, στην περίπτωση που μας ενδιαφέρει η απόκλιση όλων των χαρακτηριστικών του προσώπου (όλων των στοιχείων μιας γραμμής), επιτυγχάνεται αυξάνοντας το μετρητή τιμών vcnt κατά ένα για κάθε φορά που ικανοποιείται η συνθήκη, δηλαδή όταν το στοιχείο που εξετάζουμε είναι ένα από αυτά που μας ενδιαφέρουν. Αν η μεταβλητή vcnt πάρει την τιμή 44, όλα τα στοιχεία της τρέχουσας γραμμής του πίνακα vals διαφέρουν κατά ποσοστό μεγαλύτερο του 10% από τα αντίστοιχα στοιχεία του πίνακα τιμών αναφοράς refv και προχωράμε στην αποθήκευση της αντίστοιχης τιμής milliseconds της χρονικής στιγμής αυτής.

Χρησιμοποιούμε την ακέραια δεκαδική μεταβλητή dt ως μετρητή για να δούμε πόσες φορές ικανοποιήθηκε η συνθήκη και επομένως πόσες χρονικές στιγμές αντιστοιχούν στις αποκλίσεις που αναζητάμε, αυξάνοντάς τον κατά ένα κάθε φορά που ικανοποιείται η συνθήκη. Πριν την αλλαγή αυτή της τιμής του όμως, τον χρησιμοποιούμε για την αποθήκευση της τιμής των milliseconds της τρέχουσας χρονικής στιγμής στο νέο μονοδιάστατο πίνακα devtimes (devtimes[dt] = msecs[i]). Ο πίνακας devtimes αρχικά λαμβάνει το μέγιστο δυνατό του μέγεθος, το οποίο είναι ίσο με το μέγεθος tsz του πίνακα msecs. Το πραγματικό μέγεθος του devtimes είναι ίσο με την τιμή που θα λάβει η μεταβλητή dt μετά την εκτέλεση του βρόχου και είναι πιθανόν να είναι μικρότερο από το tsz. Στη συνέχεια αποθηκεύουμε την τιμή του μετρητή i, ο οποίος δείχνει στην τρέχουσα γραμμή του διδιάστατου πίνακα vals ή αντίστοιχα στο τρέχον στοιχείο του πίνακα milliseconds msecs, στο μονοδιάστατο πίνακα ακεραίων dptr, ο οποίος θα χρησιμοποιηθεί στη συνέχεια για ευκολότερη έυρεση των γραμμών στοιχείων του πίνακα vals που ικανοποιούν τις παραμέτρους αναζήτησης και αποθήκευση των τιμών αυτών σε νέο πίνακα. Με αυτόν τον τρόπο γίνεται ευκολότερη και η εκτίμηση του μεγέθους του πίνακα αυτού, καθώς και η δέσμευση μνήμης για αυτόν, όπως θα δούμε στη συνέχεια. Πριν την επόμενη εκτέλεση του εμφωλευμένου βρόχου, ο μετρητής dt αυξάνεται κατά ένα και ο μετρητής τιμών vcnt μηδενίζεται για εξέταση της επόμενης γραμμής. Επιπροσθέτως, χρησιμοποιώντας τη συνάρτηση fprintf με το δείκτη ροής wfp, εξάγουμε τις τιμές των χρονικών στιγμών που εντοπίσαμε στο αρχείο κειμένου εξόδου, του οποίου το όνομα δίνεται από το χρήστη ως παράμετρος εισόδου.

Ο εμφωλευμένος βρόχος τερματίζει όταν εξεταστούν όλα τα σημεία μιας γραμμής, επιτρέποντας έτσι στον εξωτερικό βρόχο να ξεκινήσει την επόμενη εκτέλεσή του για την εξέταση της επόμενης γραμμής του πίνακα vals. Όταν επεξεργαστούμε και την τελευταία γραμμή του vals, τερματίζει και ο αρχικός βρόχος for.

Η διαδικασία που παρουσιάστηκε είναι ιδιαίτερα χρήσιμη αν μας ενδιαφέρει η απόκλιση που ψάχνουμε να εμφανίζεται για όλα τα στοιχεία μιας γραμμής του διδιάστατου πίνακα vals, επομένως για όλα τα χαρακτηριστικά του προσώπου του παίκτη τη δεδομένη χρονική στιγμή ή στο συγκεκριμένο καρέ. Με αυτόν τον τρόπο εντοπίζουμε τα καρέ όπου ο παίκτης πιθανότατα θα εκτελεί πιο απότομες κινήσεις, οι οποίες όμως θα αφορούν περισσότερο τη συνολική κίνηση του κεφαλιού παρά των επιμέρους χαρακτηριστικών του προσώπου. Στη συγκεκριμένη εργασία μας ενδιαφέρει περισσότερο η μελέτη των καρέ όπου έστω και ένα από τα σημεία του προσώπου που εξετάζονται, έστω προς μια μόνο κατεύθυνση, παρουσιάζει μεγάλη απόκλιση από την ουδέτερη κατάστασή του. Αυτός ο σκοπός επιτυγχάνεται ως εξής:

```
bool nextl;
```

```
for(i=0; i<lines; i++){
	j = 0;
	nextl = false;
	while ((j<44)&&(!nextl)){
		if ((vals[i][j]>refv[j]*1.1)||(vals[i][j]<refv[j]*0.9)){
			nextl = true;
			devtimes[dt] = msecs[i]; dptr[dt] = i; dt++;
			fprintf(wfp, "millisecs[%d] = %d\n", i, msecs[i]);
		}
		j++;
	}
}
```

Ορίζουμε τη λογική μεταβλητή τύπου boolean ως μια σημαία για την εκτέλεση του νέου εμφωλευμένου βρόχου while. Ο εξωτερικός βρόχος for για την εξέταση των γραμμών παραμένει ίδιος και τερματίζει όταν ο μετρητής i πάρει την τιμή [lines] και επομένως έχουν διαβαστεί όλες οι γραμμές του πίνακα vals. Ο εμφωλευμένος βρόχος είναι πλέον ένας βρόχος while, ο τερματισμός του οποίου εξαρτάται από το μετρητή των στηλών/στοιχείων κάθε γραμμής και τη μεταβλητή nextl. Χρησιμοποιείται πάλι ο λογικός τελεστής "&&" (and), έτσι ώστε ο βρόχος να τερματίζει όταν ο μετρητής στηλών j γίνει ίσος με 43 και όταν η μεταβλητή nextl πάρει την τιμή true. Πριν την εκτέλεση κάθε επανάληψης του βρόχου while, μηδενίζεται ο δείκτης j και η τιμή της μεταβλητής nextl τίθεται ίση με false.

Μέσα στο βρόχο while εκτελείται η ίδια συνθήκη σύγκρισης με την προηγούμενη περίπτωση ((vals[i][j]>refv[j]*1.1) || (vals[i][j]<refv*0.9)) και σε περίπτωση που είναι αληθής σημαίνει ότι βρέθηκε μια τιμή των χαρακτηριστικών του προσώπου που έχουν αποθηκευθεί στο διδιάστατο πίνακα vals η οποία είναι μικρότερη του 90% ή μεγαλύτερη του 110% του αντίστοιχου στοιχείου του πίνακα τιμών αναφοράς refv. Στη συγκεκριμένη περίπτωση μας αρκεί μόνο μια τιμή τέτοιας απόκλισης, θέλουμε επομένως ο εμφωλευμένος βρόχος να τερματίζει όταν αυτή η τιμή βρεθεί και το πρόγραμμα να προχωράει σε επεξεργασία της επόμενης γραμμής. Αυτό το επιτυγχάνουμε με τη χρήση της νέας λογικής μεταβλητής nextl, στην οποία και αναθέτουμε την τιμή true όταν η συνθήκη των ανισώσεων ικανοποιείται. Σταματάμε με αυτόν τον τρόπο την επόμενη εκτέλεση του βρόχου while και ο εξωτερικός βρόχος for προχωρά στην επόμενη γραμμή του πίνακα vals. Οι διαδικασίες αποθήκευσης των χρονικών στιγμών που αντιστοιχούν στις αποκλίσεις που αναζητάμε, καθώς και των δεικτών που θα χρησιμοποιήσουμε για την αποθήκευση των τιμών των χαρακτηριστικών μεγάλης απόκλισης, πραγματοποιούνται με τον ίδιο τρόπο, όπως στην προηγούμενη περίπτωση. Αν η συνθήκη του if δεν ικανοποείται, η μόνη λειτουργία που εκτελείται είναι η αύξηση του δείκτη j κατά ένα, έτσι ώστε να προχωρήσουμε στην εξέταση του επόμενου στοιχείου της γραμμής του πίνακα vals. Τέλος, σε κάθε εκτέλεση του εξωτερικού βρόχου for, όπως αναφέραμε, μηδενίζεται ο δείκτης j έτσι ώστε να μην προκύψει κάποιο σφάλμα στην εξέταση των στοιχείων κάθε γραμμής και ο βρόχος while να εξετάζει από την αρχή όλα τα στοιχεία αυτά όταν χρειάζεται. Επίσης δίνεται στη μεταβλητή nextl η τιμή false, έτσι ώστε να αποφευχθούν πιθανά σφάλματα από προηγούμενη ανάθεση τιμής true σε αυτήν. Χωρίς το μηδενισμό της nextl, σε περίπτωση που ικανοποείται η συνθήκη if στον εμφωλευμένο βρόχο, ο βρόχος αυτός δε θα εκτελεστεί ξανά και κατά πάσα πιθανότητα δε θα έχουν εξεταστεί όλες οι γραμμές (το πιο πιθανόν μάλιστα θα ήταν να εκτελεστεί μια μικρή μειοψηφεία τους). Αυτό σημαίνει ότι η πρώτη τιμή μεγάλης απόκλισης που θα εντοπιστεί θα τερματίσει και την εκτέλεση του εξωτερικού βρόχου for, χωρίς να γίνει κάποια άλλη λειτουργία.

Δέσμευση Μνήμης και Αποθήκευση Πίνακα Τιμών Μεγάλης Απόκλισης

Δεσμεύεται μνήμη για το διδιάστατο πίνακα dval:

```
if ((dval = (float **)malloc(sizeof(float *)*dt))==NULL){
	printf("Out of memory - vals/rows\n");
	exit(-2);
}
for(i = 0; i < dt; i++){
	if((dval[i] = (float*)malloc(sizeof(float)*arr))==NULL){
		printf("Out of memory - vals/columns\n");
		exit(-2);
	}
}
```

Όπως στη δέσμευση μνήμης για τον πίνακα vals, που περιέχει όλες τις τιμές χαρακτηριστικών προσώπου που διαβάστηκαν από το αρχείο κειμένου εισόδου, για τη διαδικασία δέσμευσης μνήμης για τον πίνακα dval χρησιμοποιήθηκε η συνάρτηση της C malloc. Ο πίνακας dval ορίζεται για την αποθήκευση των τιμών του πίνακα vals που περιλαμβάνονται στις γραμμές του αρχείου κειμένου των χρονικών στιγμών του πίνακα devtimes, στις οποίες παρατηρήθηκε για τουλάχιστον ένα στοιχείο του vals, μεγαλύτερη του 10% απόκλιση από το αντίστοιχο στοιχείο του πίνακα αναφοράς refv. Ο πίνακας dval θα περιέχει τιμές τύπου float και θα έχει δύο διαστάσεις. Ο αριθμός των γραμμών του θα είναι ίσος με την τιμή της μεταβλητής dt, στην οποία βρίσκεται αποθηκευμένο το πλήθος των χρονικών στιγμών (milliseconds) στις γραμμές των οποίων παρατηρήθηκαν οι αποκλίσεις. Η μέγιστη τιμή που μπορεί να πάρει ο αριθμός των γραμμών είναι ίση με την τιμή tsz, δηλαδή με τον αριθμό γραμμών του vals και τον αριθμό στοιχείων του msecs. Ο αριθμός των στηλών του πίνακα θα είναι ίσος με arr (44) και ίδιος με τον αριθμό των στηλών του vals, καθώς μας ενδιαφέρουν όλες οι τιμές κάθε γραμμής/καρέ/χρονικής στιγμής στην οποία παρατηρήθηκε απόκλιση μεγαλύτερη του 10%, έστω και για μία μόνο τιμή της.

Ξέροντας πλέον τα μεγέθη των δύο διαστάσεων του πίνακα dval, προχωράμε στη δέσμευση μνήμης για αυτόν με [dt+1] κλήσεις της συνάρτησης malloc. Η διαδικασία που ακολουθείται είναι σχεδόν η ίδια με τη διαδικασία δέσμευσης μνήμης για τον διδιάστατο πίνακα vals, με τη διαφορά ότι για τις γραμμές δεσμεύεται μνήμη μεγέθους (sizeof(float *)*dt) και για τις στήλες, ο αντίστοιχος βρόχος for εκτελείται dt φορές.

Βρίσκουμε και αποθηκεύουμε τα στοιχεία των γραμμών μεγάλης απόκλισης:

```
for(i=0; i<dt; i++){
	for(j=0; j<44; j++){
		dval[i][j] = vals[dptr[i]][j];
	}
}
```

Χρησιμοποιούμε δύο βρόχους for, έναν εξωτερικό και έναν εμφωλευμένο, για να αποθηκεύσουμε στον πίνακα dval τις τιμές που περιλαμβάνονται στις γραμμές όπου εντοπίστηκε μεγαλύτερη του 10% απόκλιση. Για την αποθήκευση όλων των στοιχείων αυτών, ο εξωτερικός βρόχος εκτελείται dt φορές (μία για κάθε γραμμή του dval) και ο εσωτερικός 44 (μία για κάθε στήλη του dval). Οι γραμμές στις οποίες παρατηρήθηκε τουλάχιστον μια τιμή μεγάλης απόκλισης εντοπίζεται χάρη στο μονοδιάστατο πίνακα ακεραίων dptr, στον οποίο αποθηκεύσαμε νωρίτερα τις τιμές του μετρητή i που

δείχνουν στις γραμμές αυτές. Για καθεμία από αυτές αποθηκεύονται στον πίνακα dval οι 44 τιμές της γραμμής στον εμφωλευμένο βρόχο (dval[i][j] = vals[dptr[i]][j]) και ο εξωτερικός βρόχος τερματίζει όταν πάρουν τις τιμές τους και οι dt γραμμές.

Εξαγωγή Χαρακτηριστικών Καρέ και Απεικόνιση Σημείων Προσώπου

Έχουμε πλέον διαθέσιμες τις τιμές των χρονικών στιγμών σε milliseconds και των χαρακτηριστικών του προσώπου παίκτη όπου παρουσιάστηκε η απόκλισης που μας ενδιαφέρει, στους πίνακες devtimes και dval αντίστοιχα. Βάσει αυτών προχωράμε στην εξαγωγή των καρέ που τους αντιστοιχούν ως αρχεία εικόνας και την απεικόνιση σε αυτά των σημείων του προσώπου κάνοντας χρήση των κανονικοποιημένων συντεταγμένων τους:

```
for(i=0; i<dt; i++){
	cap.set(CV_CAP_PROP_POS_MSEC, (int)devtimes[i]);
	cap.read (frame);
	if(frame.empty()) break;
	char filename[80];
	for(j=6; j<43; j++){
		y = dval[i][j]; x = dval[i][j+1]; j++;//*1.68
		circle(frame, Point(x,y),3, Scalar(255,0,0),CV_FILLED, 8,0);
	}
	sprintf(filename,"dFrame_%d.jpg",i);
	imwrite(filename, frame);

}
```

Η διαδικασία πραγματοποιείται με χρήση κάποιων συναρτήσεων και κλάσεων των βιβλιοθηκών όρασης υπολογιστών ανοιχτού κώδικα OpenCV. Χρησιμοποιείται κυρίως το στιγμιότυπο cap της κλάσης VideoCapture και το στιγμιότυπο frame της κλάσης Mat της OpenCV, όπως αυτά ορίστηκαν προηγουμένως. Πιο συγκεκριμένα, για το στιγμιότυπο cap χρησιμοποιήθηκαν οι εξής μέθοδοι της κλάσης VideoCapture:

– virtual bool cv::VideoCapture::set (int propid, double value): Η μέθοδος set ορίζει μια ιδιότητα της κλάσης VideoCapture. Χρησιμοποιεί τις παραμέτρους propld και value. Η παράμετρος value είναι τύπου double και περιέχει την τιμή που θέλουμε να δώσουμε στην ιδιότητα που προσδιορίζει η ακέραια παράμετρος propid. Η ιδιότητα αυτή μπορεί να είναι μια από τις ακόλουθες:

- **CAP_PROP_POS_MSEC** Τρέχουσα θέση του αρχείου βίντεο σε milliseconds.
- **CAP_PROP_POS_FRAMES** Βασισμένος στο μηδέν δείκτης του επόμενου καρέ που θα αποκωδικοποιηθεί ή θα καταγραφεί.
- **CAP_PROP_POS_AVI_RATIO** Σχετική θέση του αρχείου βίντεο: 0 – αρχή του βίντεο, 1 – τέλος του βίντεο.
- **CAP_PROP_FRAME_WIDTH** Πλάτος των καρέ στη ροή βίντεο.
- **CAP_PROP_FRAME_HEIGHT** Ύψος των καρέ στη ροή βίντεο.
- **CAP_PROP_FPS** Ρυθμός καρέ (Frame rate).
- **CAP_PROP_FOURCC** Κώδικας τεσσάρων χαρακτήρων του κωδικοποιητή.
- **CAP_PROP_FRAME_COUNT** Αριθμός των καρέ στο αρχείο βίντεο.
- **CAP_PROP_FORMAT** Μορφή των αντικειμένων κλάσης Mat που επιστρέφει

η μέθοδος retrieve().

- **CAP_PROP_MODE** Τιμή συγκεκριμένη για το παρασκήνιο (backend) που δείχνει τον τρέχοντα τρόπο καταγραφής.
- **CAP_PROP_BRIGHTNESS** Φωτεινότητα της εικόνας (μόνο για κάμερες).
- **CAP_PROP_CONTRAST** Αντίθεση της εικόνας (μόνο για κάμερες).
- **CAP_PROP_SATURATION** Κορεσμός της εικόνας (μόνο για κάμερες).
- **CAP_PROP_HUE** Απόχρωση της εικόνας (μόνο για κάμερες).
- **CAP_PROP_GAIN** Κέρδος της εικόνας (μόνο για κάμερες).
- **CAP_PROP_EXPOSURE** Έκθεση (μόνο για κάμερες).
- **CAP_PROP_CONVERT_RGB** Λογικές (Boolean) σημαίες που δείχνουν αν οι εικόνες πρέπει να μετασχηματιστούν σε πρότυπο χρώματος RGB (κόκκινο-πράσινο-μπλε).
- **CAP_PROP_WHITE_BALANCE** Δεν υποστηρίζεται προς το παρόν.
- **CAP_PROP_RECTIFICATION** Σημαία διόρθωσης για στέρεο κάμερες (υποστηρίζεται μόνο από DC1394 v 2.x backend προς το παρόν).

– virtual bool cv::VideoCapture::read (OutputArray image): Η μέθοδος read της κλάσης VideoCapture συλλαμβάνει, αποκωδικοποιεί και επιστρέφει το επόμενο καρέ του βίντεο. Οι μέθοδοι και συναρτήσεις συνδυάζουν τις VideoCapture::grab και VideoCapture::retrieve σε μια κλήση. Η μέθοδος grab συλλαμβάνει το επόμενο καρέ από αρχείο βίντεο ή εξοπλισμό καταγραφής και η μέθοδος retrieve αποκωδικοποιεί και επιστρέφει το καρέ του βίντεο. Η read είναι η πιο βολική μέθοδος για ανάγνωση αρχείων βίντεο ή καταγραφή δεδομένων από κάμερα και για αποκωδικοποίηση και επιστροφή μόνο του καρέ που συνέλαβε. Αν δεν έχει πάρει κάποιο καρέ (η κάμερα αποσυνδέθηκε ή δεν υπάρχουν άλλα καρέ στο αρχείο βίντεο), η μέθοδος επιστρέφει false και οι συναρτήσεις επιστρέφουν μηδενικό (NULL) δείκτη.

Σε αυτό το κομμάτι της εφαρμογής dFrame οι μέθοδοι set και read της κλάσης VideoCapture χρησιμοποιήθηκαν για κάθε γραμμή του πίνακα dval (και επομένως κάθε εκτέλεση του εξωτερικού βρόχου for) ως εξής:

```
cap.set(CV_CAP_PROP_POS_FRAMES, (int)dptr[i]);
cap.read (frame);
```

Χρησιμοποιώντας τη set, η τιμή της ιδιότητας CV_CAP_PROP_POS_FRAMES του στιγμιοτύπου cap της VideoCapture τίθεται ίση με την τρέχουσα τιμή του πίνακα dptr των καρέ μεγάλης απόκλισης. Με αυτόν τον τρόπο, η τρέχουσα θέση του αρχείου βίντεο εισόδου είναι πλέον ίση, για κάθε επανάληψη του βρόχου, με την αντίστοιχο καρέ της χρονικής στιγμής όπου παρουσιάζεται απόκλιση. Στη συνέχεια με κλήση της μεθόδου read στο στιγμιότυπο cap, θα διαβαστεί και θα αποκωδικοποιηθεί το καρέ που βρίσκεται στη θέση/χρονική στιγμή που όρισε η κλήση της set. Η read επιστρέφει το καρέ αυτό στο στιγμιότυπο frame της κλάσης Mat, όπως αυτό ορίστηκε παραπάνω.

Έχοντας αποθηκεύσει το τρέχον καρέ, χρησιμοποιούμε τη μέθοδο empty (bool cv::Mat::empty () const) στο στιγμιότυπο frame της κλάσης Mat της OpenCV για να ελέγξουμε αν το frame περιέχει κάποια τιμή:

```
if(frame.empty()) break;
char filename[80];
```

Αν ο πίνακας του frame δεν περιέχει στοιχεία, η μέθοδος empty επιστρέφει με την τιμή true. Αν η empty επιστρέψει true στην κλήση της για το στιγμιότυπο frame, χρησιμοποιούμε την εντολή break για άμεσο τερματισμό του βρόχου. Ορίζουμε επίσης ένα νέο μονοδιάστατο πίνακα χαρακτήρων με όνομα filename, μεγέθους 80, στον οποίο θα αποθηκεύεται το όνομα κάθε αρχείου εικόνας που εξάγουμε. Τα αρχεία εικόνας αυτά θα απεικονίζουν στα καρέ που βρέθηκαν τα σημεία του προσώπου του παίκτη που εξετάζονται.

Ο εμφωλευμένος βρόχος for χρησιμοποιείται για την απεικόνιση των 19 σημείων του προσώπου του παίκτη στα καρέ που εξάγουμε:

```
for(j=6; j<43; j++){
      y = dval[i][j]; x = dval[i][j+1]; j++;//*1.68
      circle(frame, Point(x,y),3, Scalar(255,0,0),CV_FILLED, 8,0);
}
```

Ο βρόχος αυτός βάσει των συνθηκών της εντολής for θα πρέπει να εκτελείται 38 φορές, καθώς τα έξι πρώτα στοιχεία κάθε γραμμής του πίνακα τιμών χαρακτηριστικών προσώπων dval δεν αποτελούν κανονικοποιημένες συντεταγμένες σημείων στο πρόσωπο του παίκτη αλλά δείκτες/σημαίες κατάστασης προσοχής και αγνοούνται στην απεικόνιση. Στην πραγματικότητα ο βρόχος εκτελείται 19 φορές (μία φορά για κάθε σημείο του προσώπου) καθώς ο μετρητής j αυξάνεται και εσωτερικά. Η τιμή dval[i][j], δηλαδή το τρέχον σημείο του πίνακα dval είναι η τιμή των συντεταγμένων για τον κάθετο άξονα y και αποθηκεύεται στη μεταβλητή y. Η τιμή των συντεταγμένων για τον οριζόντιο άξονα x είναι η επόμενη τιμή της τρέχουσας γραμμής του πίνακα (dval[i][j+1]) και ανατίθεται στη μεταβλητή x. Στη συνέχεια ο μετρητής j αυξάνεται κατά ένα για να προχωρήσουμε στις συντεταγμένες του επόμενου σημείου (δηλαδή στα δύο επόμενα στοιχεία της τρέχουσας γραμμής) στην επόμενη εκτέλεση του εμφωλευμένου βρόχου. Οι μεταβλητές x και y θα χρησιμοποιηθούν από τη συνάρτηση circle της OpenCV για απεικόνιση των σημείων.

Η συνάρτηση circle (void circle(Mat& img, Point center, int radius, const Scalar& color, int thickness=1, int lineType=8, int shift=0) είναι μια από τις συναρτήσεις σχεδιασμού των βιβλιοθηκών όρασης υπολογιστών OpenCV, οι οποίες δουλεύουν με εικόνες αυθαίρετου βάθους. Τα όρια των σχημάτων μπορούν να τεθούν με antialiasing (υλοποιημένο προς το παρόν μόνο για εικόνες 8-bit). Όλες οι συναρτήσεις αυτές χρησιμοποιούν την παράμετρο color (χρώμα) που χρησιμοποιεί μια τιμή RGB (η οποία μπορεί να κατασκευαστεί με το CV_RGB ή τον κατασκευαστή Scalar_) για έγχρωμες εικόνες και φωτεινότητα για ασπρόμαυρες εικόνες. Για τις έγχρωμες εικόνες, η σειρά των καναλιών είναι συνήθως Μπλε, Πράσινο, Κόκκινο και αυτό περιμένουν οι συναρτήσεις imshow(), imread() και imwrite().

Η circle χρησιμοποιείται προφανώς για το σχεδιασμό ενός κύκλου. Η εκτέλεσή της βασίζεται στις εξής παραμέτρους:

- img: Η εικόνα (κλάσης Mat) όπου σχεδιάζεται ο κύκλος,
- center: Το κέντρο του κύκλου,
- radius: Η ακτίνα του κύκλου,
- color: Το χρώμα του κύκλου,
- thickness: Το πάχος του περιγράμματος του κύκλου, αν είναι θετικό. Αρνητικό πάχος σημαίνει ότι θα σχεδιαστεί ένας γεμάτος κύκλος,
- lineType: Τύπος του ορίου του κύκλου (8 για 8-συνδεδεμένη γραμμή, 4 για 4-συνδεδεμένη γραμμή, CV_AA για antialiased γραμμή),

– shift: Ο αριθμός των κλασματικών bits στις συντεταγμένες του κέντρου και στην ακτίνα του κύκλου.

Η συνάρτηση circle σχεδιάζει έναν απλό ή γεμάτο κύκλο (δίσκο) δεδομένου ενός κέντρου και μιας ακτίνας. Στη συγκεκριμένη περίπτωση, η παράμετρος img θα είναι το στιγμιότυπο frame της κλάσης Mat, δηλαδή το τρέχον καρέ μεγάλης απόκλισης που μας επιστρέφει η μέθοδος read για το στιγμιότυπο cap της κλάσης VideoCapture, που αντιστοιχεί στο αρχείο βίντεο εισόδου.

Η παράμετρος center καθορίζεται από την πρότυπη κλάση Point της OpenCV για σημεία δύο διαστάσεων που ορίζονται από τις συντεταγμένες τους x και y. Ένα στιγμιότυπο της κλάσης μπορεί να αντικατασταθεί με τις δομές CvPoint και CvPoint2D32f της C. Υπάρχει επίσης ένας τελεστής για αλλαγή τύπου (type casting) για την μετατροπή των συντεταγμένων σημείου στον τύπο που διευκρινίζεται. Η μετατροπή από συντεταγμένες κινητής υποδιαστολής σε ακέραιες συντεταγμένες γίνεται με στρογγυλοποίηση. Συνήθως η μετατροπή χρησιμοποιεί αυτή τη λειτουργία για καθεμία από τις συντεταγμένες. Στη συγκεκριμένη περίπτωση, το κέντρο του κύκλου θα έχει τις συντεταγμένες x και y που εντοπίσαμε στον dval.

Στην παράμετρο radius (ακτίνα του κύκλου) δίνεται η τιμή 3.

Η παράμετρος color καθορίζεται από την πρότυπη κλάση Scalar της OpenCV. Ο κύκλος που αντιπροσωπεύει κάθε σημείο θα έχει μπλε χρώμα το οποίο ορίζεται σαν τιμή RGB με τη Scalar(255, 0, 0), δίνοντας δηλαδή τη μέγιστη τιμή (255) στο μπλε χρώμα και μηδενικές τιμές για τα πράσινο και κόκκινο.

Στην παράμετρο thickness (πάχος) δίνεται η αρνητική τιμή CV_FILLED, έτσι ώστε να χρωματιστεί και το εσωτερικό του κύκλου.

Η παράμετρος lineType παίρνει την τιμή 8 για 8-συνδεδεμένη γραμμή.

Η παράμετρος shift παίρνει την τιμή 0.

Ο εμφωλευμένος βρόχος εκτελείται 19 φορές για την απεικόνιση όλων των σημείων της τρέχουσας γραμμής. Μετά τον τερματισμό του εξάγουμε το καρέ με τα 19 σημεία τοποθετημένα σε αυτό σαν αρχείο εικόνας και την ονομασία του:

```
sprintf(filename,"dFrame_%d.jpg",i);
imwrite(filename, frame);
```

Χρησιμοποιούμε τη συνάρτηση sprintf (int sprintf (char * str, const char * format, …)) της C για να αναθέσουμε έναν αύξοντα αριθμό στο όνομα κάθε frame. Η συνάρτηση αυτή συνθέτει ένα string με το ίδιο κείμενο που θα τυπωνόταν αν το format (μορφή) χρησιμοποιείτο στην printf, αλλά αντί να εκτυπωθεί, το περιεχόμενο αποθηκεύεται σαν string C στον buffer που δείχνει ο str. Το μέγεθος του buffer πρέπει να είναι αρκετά μεγάλο για να περιέχει ολόκληρο το string που προκύπτει (η snprintf είναι μια πιο ασφαλής εκδοχή). Μετά το περιεχόμενο προσαρτάται αυτόματα ένας μηδενικός (null) χαρακτήρας τερματισμού. Μετά την παράμετρο format, η συνάρτηση περιμένει τουλάχιστον τόσες παραμέτρους όσε χρειάζονται για το format. Εδώ η sprintf χρησιμοποιείται με τον πίνανκα χαρακτήρων (string) filename, το format %d για ακεραίους δεκαδικούς και το μετρητή i ως πρόσθετη παράμετρο, με τέτοιον τρόπο ώστε τα ονόματα των αρχείων εικόνων που προκύπτουν να έχουν τη μορφή dFrame_0.jpg, dFrame_1.jpg, dFrame_2.jpg κ.ο.κ. Το πλήθος των ονομάτων που προκύπτουν είναι ίσο με dt.

Χρησιμοποιούμε τη συνάρτηση imwrite της OpenCV (bool imwrite(const string& filename, InputArray img, const vector<int>&params=vector<int>()). Η συνάρτηση imwrite αποθηκεύει μια

εικόνα σε ένα καθορισμένο αρχείο. Οι παράμετροι που χρησιμοποιεί είναι:

- filename: το όνομα του αρχείου,
- image: η εικόνα που θέλουμε να αποθηκεύσουμε,
- params: παράμετροι σχετικοί με τη μορφή (format), όπως παράμετρος ποιότητας για JPEG (CV_IMWRITE_JPEG_QUALITY) ή επιπέδου συμπίεσης για PNG (CV_IMWRITE_PNG_COMPRESSION) κ.α.

Η συνάρτηση imwrite αποθηκεύει την εικόνα στο καθορισμένο αρχείο. Η μορφή της εικόνας επιλέγεται βασισμένη στην επέκταση filename. Μόνο εικόνες 8-bit (ή 16-bit χωρίς πρόσημο (CV_16U) στην περίπτωση των PNG, JPEG 2000 και TIFF) μονού καναλιού ή τριπλού καναλιού (με σειρά καναλιών BGR) μπορούν να αποθηκευθούν χρησιμοποιώντας αυτήν τη συνάρτηση. Αν η σειρά μορφής, βάθους ή καναλιού είναι διαφορετική, χρησιμοποιείται η Mat::convertTo() και η cvtColor() για την μετατροπή πριν την αποθήκευση. Εναλλακτικά χρησιμοποιούνται οι καθολικές συναρτήσεις εισόδου/εξόδου FileStorage για αποθήκευση της εικόνας σε μορφή XML ή YAML. Είναι πιθανή η αποθήκευση εικόνων PNG με κανάλι άλφα χρησιμοποιώντας αυτήν τη συνάρτησης. Για να το επιτύχουμε αυτό, πρέπει να δημιουργήσουμε 8-bit (ή 16-bit) εικόνα τετραπλού καναλιού εικόνας BGRA, όπου το κανάλι άλφα είναι τελευταίο. Εντελώς διαφανή εικονοστοιχεία πρέπει να έχουν ορίσει το άλφα ως μηδέν, πλήρως αδιαφανή στοιχεία πρέπει να έχουν ορίσει την τιμή του άλφα στο 255/65535.

Σε αυτό το κομμάτι της εφαρμογής dFrame, απλά καλείται ως imwrite(filename,frame) αποθηκεύοντας με μορφή JPEG τα καρέ που εντοπίζονται σε κάθε επανάληψη του εξωτερικού βρόχου for και βρίσκονται στα αντίστοιχα στιγμιότυπα frame της Mat, στους πίνακες filename. Ο βρόχος εκτελείται μία φορά για κάθε γραμμή του dval, δηλαδή dt φορές.

Η εφαρμογή dFrame έχει σχεδόν τελειώσει και αποδεσμεύουμε τη μνήμη που δεσμεύσαμε για τους πίνακες και τους δείκτες ροής αρχείων:

```
free(vals); free(msecs); free(string);
fclose(fp); printf("close\n");
//fclose(wfp); printf("wfp\n");
//fclose(yaf); printf("yaf\n");

return 0;
```

Η συνάρτηση της C void free (void* ptr) χρησιμοποιείται για αποδέσμευση μπλοκ μνήμης. Αποδεσμεύεται ένα μπλοκ μνήμης το οποίο είχε δεσμευτεί νωρίτερα από μια κλήση στη malloc, την calloc ή τη realloc, και καθίσταται διαθέσιμο για περεταίρω δεσμεύσεις μνήμης. Αν ο δείκτης ptr δε δείχνει σε ένα μπλοκ μνήμης που δεσμεύτηκε με τις συναρτήσεις αυτές, προκαλεί απροσδιόριστη συμπεριφορά (undefined behavior). Αν ο δείκτης ptr είναι μηδενικός (null), η συνάρτηση δεν κάνει κάτι. Η συνάρτηση αυτή δεν αλλάζει την τιμή του ίδιου του ptr, ο οποίος επομένως συνεχίζει να δείχνει στην ίδια, άκυρη πλέον, τοποθεσία. Η free χρησιμοποιείται για αποδέσμευση της μνήμης των πινάκων vals, msecs και string.

Η συνάρτηση της C int fclose (FILE * stream) χρησιμοποιείται για κλείσιμο αρχείων. Κλείνει το αρχείο που έχει συσχετιστεί με τη ροή stream και το αποσυνδέει. Όλοι οι εσωτερικοί buffers που είχαν συσχετιστεί με το αρχείο διαχωρίζονται από αυτό και σβήνονται: το περιεχόμενο οποιουδήποτε buffer εξόδου χωρίς εγγραφές εγγράφεται και το περιεχόμενο οποιουδήποτε μη αναγνωσμένου buffer εισόδου απορρίπτεται. Ακόμα κι αν η κλήση αποτύχει, η ροή που δόθηκε ως παράμετρος δε θα

συσχετίζεται πλέον με το αρχείο ή τους buffers του. Αν η ροή κλείσει επιτυχώς, επιστρέφεται μια μηδενική τιμή. Αν η συνάρτηση αποτύχει, επιστρέφει EOF. Η fclose χρησιμοποιείται για κλείσιμο του αρχείου εισόδου μέσω της ροής fp.

Πριν το τέλος της main, χρησιμοποιούμε την εντολή return 0; έτσι ώστε αν η κλήση της main επιτύχει στις παραπάνω διαδικασίες επιστρέφεται η τιμή μηδέν, ενώ αν αποτύχει θα επιστρέψει κάποια αρνητική τιμή.

Τελική Περιγραφή

Συνοπτικά, το πρόγραμμα dFrame υλοποιήθηκε στο ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse CDT για εφαρμογές σε γλώσσες C/C++, με χρήση βιβλιοθηκών όρασης υπολογιστών ανοιχτού κώδικα OpenCV. Το πρόγραμμα δέχεται τρεις παραμέτρους εισόδου, ένα αρχείο κειμένου εισόδου, ένα δεύτερο αρχείο κειμένου που δημιουργείται για αποτελέσματα και ένα αρχείο βίντεο εισόδου, και υλοποιήθηκε ως εξής:

- μετά τις αρχικές αναθέσεις και αρχικοποιήσεις, το πρόγραμμα δεσμεύει μνήμη για τους πίνακες που θα χρησιμοποιήσει,
- ορίζει έναν πίνακα τιμών αναφοράς,
- διαβάζει το αρχείο κειμένου εισόδου και αποθηκεύει σε δύο πίνακες, το μονοδιάστατο πίνακα χρόνου msecs και το διδιάστατο πίνακα τιμών val τα δεδομένα του αρχείου αυτού,
- συγκρίνει τις τιμές του πίνακα val με τις αντίστοιχες τιμές του πίνακα αναφοράς και αν βρεθεί απόκλιση έστω και μιας από αυτές για κάθε γραμμή, αποθηκεύει την αντίστοιχη τιμή του πίνακα χρόνου msec στο νέο μονοδιάστατο πίνακα devtimes, και εξάγει τις χρονικές τιμές αυτές στο αρχείο κειμένου αποτελεσμάτων,
- ορίζει το διδιάστατο πίνακα dval με τις τιμές του val που αντιστοιχούν στις χρονικές στιγμές του devtimes,
- χρησιμοποιεί την OpenCV και τους πίνακες devtimes και dval για την εξαγωγή από το αρχειο βίντεο εισόδου των καρέ που αντιστοιχούν στις χρονικές στιγμές του devtimes και για την απεικόνιση των σημείων, των οποίων οι συντεταγμένες δίνονται από τον dval, στα νέα αρχεία εικόνας που δημιουργεί,
- αποδεσμεύει τη μνήμη και επιστρέφει την τιμή 0.

Παρουσιάζεται ο πηγαίος κώδικας της dFrame στο σύνολο του:

```
/*
 * main.cpp
 *
 *  Created on: Feb 18, 2016
 *      Author: Slartibartfast
 */
```

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>

using namespace cv;

int main(int argc, char *argv[])
{
	/*Variables*/
	int i=0, lines=0, lns=0, dt=0, x, y;
	int arr, tsz, j;
	int ns, vcnt;
	int n, n1, total_n=0;//w
	char ch, *string;
	float **vals, **dval, w, refv[44];
	FILE *fpt, *fp, *wfp, *yaf;
	Mat frame;

	/*File Open & Initial Values*/
	if (argc!=4) {
		printf( "usage: %s input-filename output-filename input-video filename\n", argv[0]);
		exit(2);
	}
	if ((fpt = fopen(argv[1],"r")) == NULL){
		printf("Open infile error\n");
		exit(1);
	}

	if ((wfp = fopen(argv[2],"w")) == NULL){
		printf("Open outfile error\n");
		exit(1);
	}
```

```
if ((yaf = fopen("yafile.txt","w")) == NULL){
        printf("Open file error\n");
        exit(1);
}


VideoCapture cap(argv[3]); // open the video file for reading

if ( !cap.isOpened() )  // if not success, exit program
{
        printf("Cannot open the video file");
        exit(-1);
}

fprintf(wfp, "Milliseconds for larger than 0.1 deviation\n\n");
//--initialize reference array (TIMES ANAFORAS 2_A)
refv[0]=0.000000; refv[1]=0.000000; refv[2]=0.000000; refv[3]=0.000000; refv[4]=1.000000;
refv[5]=0.000000;  refv[6]=203.000000;  refv[7]=349.000000;  refv[8]=201.000000;
refv[9]=327.000000;
refv[10]=202.000000;  refv[11]=412.000000;  refv[12]=202.000000;  refv[13]=389.000000;
refv[14]=202.000000;
refv[15]=335.000000;  refv[16]=202.000000;  refv[17]=403.000000;  refv[18]=204.000000;
refv[19]=339.000000;
refv[20]=197.000000;  refv[21]=339.000000;  refv[22]=207.000000;  refv[23]=401.000000;
refv[24]=197.000000;
refv[25]=401.000000;  refv[26]=281.000000;  refv[27]=335.000000;  refv[28]=282.000000;
refv[29]=396.000000;
refv[30]=291.000000;  refv[31]=366.000000;  refv[32]=273.000000;  refv[33]=368.000000;
refv[34]=192.000000;
refv[35]=351.000000;  refv[36]=184.000000;  refv[37]=327.000000;  refv[38]=188.000000;
refv[39]=412.000000;
refv[40]=192.000000; refv[41]=391.000000; refv[42]=253.000000; refv[43]=370.000000;
/*TIMES ANAFORAS 1_A
 * 195.006134 350.192017 193.504654 327.062012 192.242157 413.986755 192.048447 390.145416
 * 193.858002 336.118073 193.204849 401.112854 196.838120 337.127014 189.856491 337.127014
 * 198.229507 402.066071 188.153061 404.066071 270.000000 337.323090 272.000000 397.468353
```

```
*  277.044983 366.488312 262.046753 368.709106 177.954483 349.075684 177.924179 327.033020
*   176.075928 414.031494 185.008453 390.233673 244.014008 370.886749*/

        //--count file lines

        while(!feof(fpt)) {
                ch = fgetc(fpt);
                if(ch == '\n') { lns++; }
        }
        fclose(fpt);

        tsz = lns;

        int *msecs = (int*)malloc(tsz*sizeof(int));
        int *devtimes = (int*)malloc(tsz*sizeof(int)); //xronikes stigmes megalis apoklisis
        int *dptr = (int*)malloc(tsz*sizeof(int)); //xronikes stigmes megalis apoklisis

        for (ns=0; ns<tsz; ns++)
                msecs[ns]=0;
        arr = 44;
        printf("lns = %d, tsz = %d, arr = %d\n",lns,tsz,arr);
        string = (char*) malloc (510);
        if ((vals = (float **)malloc(sizeof(float *)*tsz))==NULL){
                printf("Out of memory - vals/rows\n");
                exit(-2);
        }
        for(i = 0; i < tsz; i++){
                if((vals[i] = (float*)malloc(sizeof(float)*arr))==NULL){
                        printf("Out of memory - vals/columns\n");
                        exit(-3);
                }
        }
        fp = fopen(argv[1],"r");
        vcnt=i=0;
```

```
/*Main Loop*/
while((fgets(string,510,fp))!=NULL){
        //--counting lines to separate time values from features values
        sscanf(string, "%d %n", &msecs[lines], &n1);
        fprintf(yaf,"%d ", msecs[lines]);
        total_n = n1; //printf("n1 %d\n", n1);
        while((1 == sscanf(string2 + total_n, "%f %n", &w, &n)) && (i<44)) {
                //line scan
                total_n += n;
                vals[lines][i] = w;
                fprintf(yaf, "%f ", vals[lines][i]);
                i++;
        } fprintf(yaf,"\n");
        i = 0; total_n = 0; //vcnt = 0;
        lines++;


}
bool nextl;
for(i=0; i<lines; i++){
        j = 0;
        nextl = false;
        while ((j<44)&&(!nextl)){
                if ((vals[i][j]>refv[j]*1.1)||(vals[i][j]<refv[j]*0.9)){
                        nextl = true;
                        devtimes[dt] = msecs[i]; dptr[dt] = i; dt++;
                        fprintf(wfp, "millisecs[%d] = %d\n", i, msecs[i]);
                }
                j++;
        }
}
printf("dt = %d\n",dt);
if ((dval = (float **)malloc(sizeof(float *)*dt))==NULL){
        printf("Out of memory - vals/rows\n");
        exit(-2);
}
for(i = 0; i < dt; i++){
        if((dval[i] = (float*)malloc(sizeof(float)*arr))==NULL){
```

```
                printf("Out of memory - vals/columns\n");
                exit(-2);
            }
        }

        for(i=0; i<dt; i++){
            for(j=0; j<44; j++){
                dval[i][j] = vals[dptr[i]][j]; //printf("dval[%d,%d] = %f\n", i, j, dval[i][j] );
            }
        }


        for(i=0; i<dt; i++){
            cap.set(CV_CAP_PROP_POS_FRAMES, (int)dptr[i]);
            cap.read (frame);
            if(frame.empty()) break;
           char filename[80];
            for(j=6; j<43; j++){
                y = dval[i][j]; x = dval[i][j+1]; j++;//*1.68
                circle(frame, Point(x,y),3, Scalar(255,0,0),CV_FILLED, 8,0);
            }
           sprintf(filename,"dFrame_%d.jpg",i);
           imwrite(filename, frame);

        }
        free(vals); free(msecs); free(string);
        fclose(fp); printf("close\n");
        //fclose(wfp); printf("wfp\n");
        //fclose(yaf); printf("yaf\n");

        return 0;
}
```

Στη συνέχεια παρουσιάζονται τα αποτελέσματα της εκτέλεσης του προγράμματος dFrame.

5. Αποτελέσματα Εφαρμογής dFrame

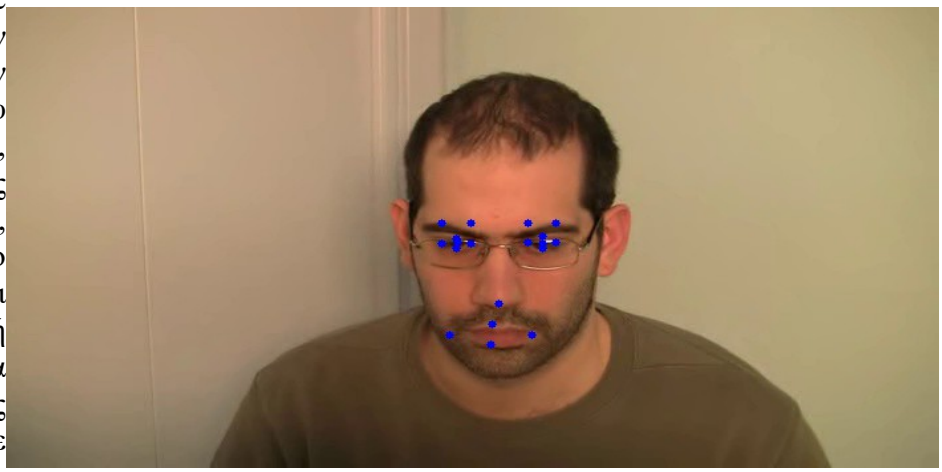
Σε αυτήν την υποπαράγραφο παρουσιάζονται κάποια αποτελέσματα της εκτέλεσης της εφαρμογής dFrame για κάποια συγκεκριμένα παραδείγματα. Η εκτελεση της dFrame παράγει αρχεία εικόνας (jpeg) έχοντας λάβει ως είσοδο το αρχείο βίντεο εισόδου μορφής avi, το οποίο περιέχει τις οπτικές αντιδράσεις ενός παίκτη για μία συνεδρία παιχνιδιού (κάθε συνεδρία αποτελείται από τρεις προσπάθειες ή "ζωές" του Μάριο).

Τα αρχεία εικόνας αυτά, είναι τα καρέ που εξάγονται από το αρχείο βίντεο εισόδου τις χρονικές στιγμές κατά τις οποίες προκύπτει απόκλιση από τις τιμές αναφοράς τουλάχιστον ενός από τα χαρακτηριστικά του προσώπου του παίκτη, μεγαλύτερη από 5.7%. Οι τιμές αναφοράς είναι προφανώς διαφορετικές για κάθε συνεδρία παιχνιδιού και ελέγχθηκαν έτσι ώστε να αντιπροσωπεύουν με τη μέγιστη δυνατή ακρίβεια τη μέση κατάσταση/κατάσταση αδρανείας του κάθε παίκτη. Ο έλεγχος των τιμών αναφοράς έγινε εξετάζοντας το αρχείο κειμένου που περιέχει τις τιμές/συντεταγμένες των σημείων του προσώπου του παίκτη δίνοντας, σε συνδυασμό με εξέταση του αντίστοιχου αρχείου βίντεο εισόδου, ιδιαίτερη σημασία στις τιμές των καρέ στα οποία ο παίκτης βρισκόταν στην αρχική του κατάσταση καθώς και στις τιμές συντεταγμένων που εμφανίστηκαν σε κάθε αρχείο κειμένου με τη μεγαλύτερη συχνότητα (βάσει του ακεραίου μέρους τους). Το ποσοστό απόκλισης 5.7% αναδείχθηκε μετά από αρκετές δοκιμές ως το πιο αποτελεσματικό για την εύρεση των χαρακτηριστικών καρέ που μας ενδιαφέρουν.

Ξεκινώντας με ποσοστό 10%, σε κάθε δοκιμή εκτελούσαμε την εφαρμογή dFrame χρησιμοποιώντας ως εισόδους τα αρχεία ενός παραδείγματος συνεδρίας παιχνιδιού "2_A". Στη συνέχεια, συγκρίναμε τα καρέ που εξήχθησαν με το αρχείο βίντεο εισόδου. Όσο τα καρέ δεν ανταποκρίνονταν πλήρως στις πιο έντονες αντιδράσεις του παίκτη, δεν περιείχαν δηλαδή τις πιο απότομες κινήσεις ή μεταβολές από την κατάσταση αδράνειας των σημείων αναφοράς, μειώναμε το ποσοστό κατά 1% και εκτελούσαμε πάλι την εφαρμογή. Στην πέμπτη δοκιμή, με ποσοστό 6%, παρατηρήσαμε ότι εξήχθησαν σχεδόν όλα τα καρέ που μας ενδιαφέρουν, ενώ στην έκτη δοκιμή με ποσοστό 5%, η εφαρμογή παρήγαγε πολλά περιττά καρέ. Ξεκινώντας πάλι από το 6%, οι επόμενες δοκιμές έγιναν εκτελώντας τη dFrame για ποσοστό που μειωνόταν κάθε φορά κατά 0.1% και έτσι καταλήξαμε στο 5.7% καθώς για 5.8% χάναμε κάποιες μικρότερες αντιδράσεις του παίκτη (οι οποίες όμως μας ενδιαφέρουν) και για 5.6% παράγονται αρκετά περιττά καρέ.

Στη συνέχεια παρουσιάζονται κάποια παραδείγματα εικόνων που παρήχθησαν από εκτέλεση της εφαρμογής με αρχεία εισόδου βίντεο και κειμένου συγκεκριμένων συνεδριών παιχνιδιού. Όπως ήταν αναμενόμενο, οι πιο έντονες από τις παρακάτω αντιδράσεις παρατηρήθηκαν κατά το συμβάν παιχνιδιού "θάνατος του Μάριο" ή όταν ο παίκτης έφτανε στο τέλος του επιπέδου, ολοκληρώνοντάς το.

Στο Καρέ 1, ο παίκτης βρίσκεται σε ουδέτερη κατάσταση (αδράνειας) και μόλις έχει ξεκινήσει τη συνεδρία παιχνιδιού Infinite Mario Bros. Οι τιμές των συντεταγμένων των χαρακτηριστικών προσώπου του παίκτη τη στιγμή αυτή, αλλά και σε άλλες αντίστοιχες καταστάσεις, εξάγονται από το αρχείο κειμένου εισόδου και ταυτίζονται με ή παρουσιάζουν αμελητέα απόκλιση από τις τιμές αναφοράς που έχουμε επιλέξει για κάθε χαρακτηριστικό προσώπου.



Καρέ 1: Ο παίκτης του παραδείγματος "2_A" σε κατάσταση αδράνειας

Από τις εμφανίσεις τιμών ουδέτερης κατάστασης και των αντιστοίχων καρέ, μπορούμε να προσδιορίσουμε τα χρονικά διαστήματα κατά τα οποία ο παίκτης βρισκόταν στην κατάσταση αυτή χωρίς να παρουσιαστούν αξιοσημείωτες μεταβολές. Συνδυάζοντας τις πληροφορίες αυτές με τα δεδομένα gameplay και περιεχομένου παιχνιδιού καταλήγουμε στο συμπέρασμα ότι ελάχιστες έως καθόλου αντιδράσεις από τον παίκτη είχαμε όσο αυτός δε δυσκολευόταν να προχωρήσει στο επίπεδο του παιχνιδιού. Αυτό κατά πάσα πιθανότητα σημαίνει μηχανική αντιμετώπιση του παιχνιδιού από τον παίκτη και πολύ χαμηλά επίπεδα εκνευρισμού και πρόκλησης. Ανάλογα με τον εκάστοτε παίκτη, την εμπειρία του και την οπτική συμπεριφορά του, το πιθανότερο είναι ότι και το επίπεδο προσήλωσης κατά τα χρονικά διαστήματα αυτά θα είναι χαμηλό.

Καρέ 2-5: Ο παίκτης του παραδείγματος "2_A" σε μεγάλη απόκλιση από την κατάσταση αδράνειας.

Στα καρέ 2 έως 5 παρουσιάζεται η πρώτη έντονη οπτική αντίδραση του παίκτη της συγκεκριμένης συνεδρίας παιχνιδιού, δηλαδή η πρώτη μεγαλύτερη του 5.7% απόκλιση των τιμών των χαρακτηριστικών του προσώπου του από τις επιλεγμένες τιμές αναφοράς. Οι διαφορές στην κατάσταση του παίκτη από την ουδέτερη κατάσταση είναι προφανείς τόσο στα μάτια του παίκτη που κλείνουν όσο και στο στόμα του, καθώς και σε μια μικρή κίνηση του κεφαλιού του. Τα συγκεκριμένα καρέ αντιστοιχούν στη χρονική στιγμή που έλαβε χώρα το συμβάν παιχνιδιού "θάνατος του Μάριο" για πρώτη φορά στη συγκεκριμένη συνεδρία παιχνιδιού. Είναι προφανές ότι την αμέσως προηγούμενη στιγμή από το συμβάν αυτό είχαμε οπτική αντίδραση του παίκτη που υποδεικνύει υψηλό επίπεδο προσήλωσης και πρόκλησης, ενώ μετά το συμβάν αυξήθηκε περισσότερο ο εκνευρισμός του παίκτη. Τα καρέ αυτά είναι χαρακτηριστικό παράδειγμα των οπτικών αντιδράσεων που μας ενδιαφέρουν περισσότερο στην εργασία αυτή με μελλοντικό στόχο την πρόβλεψη των καταστάσεων προσήλωσης, πρόκλησης και εκνευρισμού. Ενώ η ένταση των αντιδράσεων και κινήσεων μπορεί να διαφέρει από παίκτη σε παίκτη (στο συγκεκριμένο παράδειγμα οι αντιδράσεις του παίκτη δεν είναι ποτέ ιδιαίτερα έντονες), επιλέγοντας τα κατάλληλα σημεία αναφοράς και το σωστό ποσοστό απόκλισης για κάθε περίπτωση μπορούμε να εντοπίσουμε τα χαρακτηριστικά καρέ που μας ενδιαφέρουν με σχετική ευκολία.

Στο καρέ 6 ο παίκτης έχει μόλις ξεκινήσει τη δεύτερη προσπάθειά του στη συγκεκριμένη συνεδρία παιχνιδιού (με τη δεύτερη ζωή του Μάριο). Παρατηρούμε ότι το καρέ 7 (λίγο αργότερα στην ίδια προσπάθεια/ζωή) έχει ελάχιστες διαφορές από το καρέ 6. Θα μπορούσε κανείς να ισχυριστεί ότι εδώ έχουμε μια νέα ουδέτερη κατάσταση. Ο ισχυρισμός αυτός όμως δε θα ήταν ακριβής, καθώς είναι φανερές οι διαφορές ανάμεσα στα δύο αυτά καρέ και στο καρέ 1 της αρχικής κατάστασης. Τα καρέ 6-7 λαμβάνουν χώρα μετά το συμβάν "θάνατος του Μάριο" που αναφέρθηκε παραπάνω και τα επίπεδα προσήλωσης αλλά και πρόκλησης του παίκτη είναι αρκετά υψηλότερα σε σχέση με την αρχή της

συνεδρίας του παιχνιδιού.

Καρέ 8: Ο παίκτης του παραδείγματος "2_A" στο τέλος της συνεδρίας παιχνιδιού, έχοντας μόλις ολοκληρώσει το επίπεδο.

Στο καρέ 8 βλέπουμε τον παίκτη τη στιγμή που ολοκληρώνει το επίπεδο του παιχνιδιού. Παρατηρούμε μια μικρή μεταβολή στα χαρακτηριστικά του προσώπου του, κυρίως στην περιοχή των φρυδιών, η οποία κατά πάσα πιθανότητα σηματοδοτεί και την επιστροφή του σε ουδέτερη κατάσταση. Μετά τις δυσκολίες (συμβάν "θάνατος του Μάριο", δυσκολότερα κενά και τοποθεσίες εχθρών) κατά τις οποίες τα επίπεδα πρόκλησης, προσήλωσης και εκνευρισμού αυξάνονταν, ο παίκτης ηρεμεί με μια τελευταία μεταβολή από την αρχική του κατάσταση.

Τα αποτελέσματα της εφαρμογής dFrame αποτελούν άλλη μια απόδειξη ότι η παρατήρηση της οπτικής συμπεριφοράς των παικτών κατά τη διάρκεια ενός παιχνιδιού, δίνοντας ιδιαίτερη προσοχή στις πιο έντονες αντιδράσεις τους, είναι πολύ σημαντική για τη διαδικασία πρόβλεψης της συμπεριφοράς των παικτών και των τριων συναισθηματικών καταστάσεων (προσήλωση, πρόκληση, εκνευρισμός). Όπως είδαμε, οι μεγαλύτερες αποκλίσεις των χαρακτηριστικών του προσώπου από την ουδέτερη κατάσταση παίκτη αντιπροσωπεύουν τις πιο έντονες αντιδράσεις του και αντιστοιχούν σε συγκεκριμένα συμβάντα του παιχνιδιού. Το συμβάν "θάνατος του Μάριο" είναι η βασική αιτία των μεγαλυτέρων μεταβολών ενώ μικρότερες συνήθως αντιδράσεις παρουσιάζονται και κατά τη διάρκεια κάποιου δύσκολου άλματος, ίσως και σε συνδυασμό με την τοποθέτηση κάποιου εχθρού. Στα συμβάντα αυτά θα έχουμε και τις μεγαλύτερες τιμές για τη συναισθηματική κατάσταση του εκνευρισμού αλλά και της πρόκλησης.

Ενδιαφέρον παρουσιάζουν και οι συγκριτικά μικρότερες μεταβολές σε σχέση με την αρχική

κατάσταση, όπως φαίνεται στα καρέ 6 και 7. Τα καρέ αυτά μας δείχνουν την κατάσταση του παίκτη μετά τον πρώτο θάνατο του Μάριο και παρά το γεγονός ότι δεν απεικονίζουν κάποια έντονη αντίδραση, έχουν αρκετές διαφορές από την αρχική κατάσταση του παίκτη όπως αυτή φαίνεται στο καρέ 1. Στο χρονικό διάστημα αυτό, που αντιστοιχεί στη δεύτερη προσπάθεια του παίκτη, παρατηρείται μια αναμενόμενη αύξηση στην κατάσταση της προσήλωσης. Ο παίκτης ήδη έχει χάσει μια ζωή και συνειδητοποιεί ότι το παιχνίδι του παρέχει ικανοποιητικό επίπεδο πρόκλησης, επομένως αυξάνεται και η προσήλωσή του σε αυτό.

Από τα παραπάνω εύκολα συμπεραίνει κανείς ότι χρησιμοποιώντας την εφαρμογή dFrame για την εύρεση των χαρακτηριστικών καρέ έντονων μεταβολών σε συνδυασμό με τα δεδομένα περιεχομένου παιχνιδιού και gameplay, διευκολύνεται σημαντικά η διαδικασία πρόβλεψης συμπεριφοράς και συναισθηματικών καταστάσεων του παίκτη.

6. Συμπεράσματα

Στην εργασία αυτή, ασχοληθήκαμε με το πρόβλημα της διαδικασίας πρόβλεψης της συμπεριφοράς παικτών κατά τη διάρκεια κάποιου ηλεκτρονικού παιχνιδιού, βάσει των τριων συναισθηματικών καταστάσεων της προσήλωσης, της πρόκλησης και του εκνευρισμού.

Παρουσιάστηκαν αποτελέσματα προηγουμένων μελετών, τα οποία υποστηρίζουν ότι ένας συνδυασμός καταγραφής και επεξεργασίας των οπτικών αντιδράσεων του κάθε παίκτη με δεδομένα περιεχομένου παιχνιδιού και gameplay είναι ένας πολύ αποδοτικός τρόπος πρόβλεψης της συναισθηματικής κατάστασης. Επίσης παρατέθηκαν και χρησιμοποιήθηκαν τα αποτελέσματα προηγούμενης έρευνας στην οποία καταγράφηκαν οι αντιδράσεις διαφόρων παικτών σε συνεδρίες παιχνιδιού Super Mario Bros.

Αναλύθηκαν οι εφαρμογές PoseGaze και FacialFeat που αποτέλεσαν βάση για την εργασία και το κύριο αντικείμενό της, την εφαρμογή dFrame. Η PoseGaze δέχεται αρχείο βίντεο ως είσοδο και καταγράφει σε αρχείο κειμένου πληροφορίες και τιμές σχετικές με το διάνυσμα Στάσης κεφαλιού και Βλέμματος, ενώ η FacialFeat καταγράφει αντίστοιχα πληροφορίες και τιμές συντεταγμένων για 19 σημεία στο πρόσωπο του παίκτη.

Χρησιμοποιώντας βασικά εργαλεία όπως η πλατφόρμα Eclipse, οι γλώσσες προγραμματισμού C και C++ και η ανοιχτή βιβλιοθήκη όρασης υπολογιστών OpenCV, αναπτύχθηκε η εφαρμογή dFrame για την επίτευξη του βασικού στόχου της εργασίας αυτής, ο οποίος είναι η εξαγωγή χαρακτηριστικών καρέ από ακολουθίες βίντεο με παίκτες Super Mario. Η εφαρμογή dFrame χρησιμοποιεί τα αρχεία κειμένου που παράγουν οι PoseGaze και FacialFeat για τον εντοπισμό των καρέ στα οποία παρατηρούνται οι εντονότερες αντιδράσεις του παίκτη, την εξαγωγή τους από το αντίστοιχο αρχείο βίντεο και την απεικόνιση σε αυτά των χαρακτηριστικών σημείων στο πρόσωπο.

Όπως αναφέρθηκε και παραπάνω, βασικός στόχος της εφαρμογής dFrame είναι η διευκόλυνση της διαδικασίας πρόβλεψης των συναισθηματικών καταστάσεων του παίκτη. Είδαμε ότι τα χαρακτηριστικά καρέ σε συνδυασμό με τα δεδομένα gameplay μπορούν να μας δώσουν με αρκετή ακρίβεια τη συναισθηματική κατάσταση του παίκτη κατά τη διάρκεια μιας συνεδρίας και επομένως μπορούν να χρησιμοποιηθούν αποδοτικά στην πρόβλεψη συμπεριφοράς ενός παίκτη, όχι μόνο σε διδιάστατα παιχνίδια πλατφόρμας όπως το Super Mario, αλλά και σε παιχνίδια άλλων ειδών, όπως τα shooters πρώτου προσώπου (FPS). Δυσκολότερη θα ήταν η εφαρμογή της διαδικασίας σε παιχνίδια με πιο σύνθετο και αργό gameplay όπως στρατηγικής και RPGs, καθώς στις κατηγορίες αυτές θα έπρεπε να ληφθούν υπόψη πολύ περισσότεροι παράγοντες.

Τελικός σκοπός της εργασίας, καθώς και αντικείμενο πιθανής μελλοντικής μελέτης, είναι η χρήση των αποτελεσμάτων αυτών έτσι ώστε η διαδικασία πρόβλεψης συμπεριφοράς παίκτη να καθιστά δυνατή τη δυναμική παραγωγή επιπέδων. Το παιχνίδι δηλαδή, λαμβάνοντας υπόψη την προσωπική εμπειρία κάθε παίκτη θα προσαρμόζεται αυτόματα, με τέτοιο τρόπο ώστε να επιτυγχάνονται τα επιθυμητά/ιδανικά επίπεδα των συναισθηματικών καταστάσεων της πρόκλησης, προσήλωσης και του εκνευρισμού για τον καθένα.

7. Βιβλιογραφία

[1] Noor Shaker, Stylianos Asteriadis, Georgios N. Yannakakis, Member, IEEE, and Kostas Karpouzis, Senior Member, IEEE (2013). Fusing Visual and Behavioral Cues for Modeling User Experience in Games.

[2] K. Höök, “Affective loop experiences—What are they?,” in Proc. PERSUASIVE, LNCS 5033. 2008, pp. 1–12.

[3] G. N. Yannakakis, M. Maragoudakis, and J. Hallam, “Preference learning for cognitive modeling: A case study on entertainment preferences,” IEEE Trans. Sys., Man, Cyber. A, Syst., Humans, vol. 39, no. 6, pp. 1165–1175, Nov. 2009.

[4] C. Pedersen, J. Togelius, and G. N. Yannakakis, “Modeling player experience for content creation,” IEEE Trans. Comput. Intell. AI Games, vol. 2, no. 1, pp. 54–67, Mar. 2010.

[5] N. Shaker, G. N. Yannakakis, and J. Togelius, “Towards automatic personalized content generation for platform games,” in Proc. AAAI Conf. Artif. Intell. Interactive Dig. Entertainment, Oct. 2010, pp. 63–68

[6] N. Shaker, G. Yannakakis, and J. Togelius, “Digging deeper into platform game level design: Session size and sequential features,” in Proc. Eur. Conf. Applicat. Evol. Comput. (EvoApplications), 2012.

[7] J. Fürnkranz and E. Hüllermeier, “Pairwise preference learning and ranking,”in Proc. Mach. Learn. ECML, 2003, pp. 145–156.

[8] P. Sundstrm, “Exploring the affective loop,” Stockholm Univ., Stockholm, Sweden, Tech. Rep., 2005.

[9] E. Hudlicka, “Affective computing for game design,” in Proc. 4th Int. North Amer. Conf. Intell. Games Simulation, 2008, pp. 5–12

[10] H. Martinez and G. Yannakakis, “Genetic search feature selection for affective modeling: A case study on reported preferences,” in Proc. 3rd Int. Workshop Affective Interaction Natural Environments, 2010, pp. 15–20.

Ηλεκτρονικές Πηγές

[11] https://en.wikipedia.org/wiki/Eclipse_(software)

[12] https://eclipse.org/ide

[13] https://eclipse.org/cdt/

[14] http://www.ibm.com/developerworks/opensource/library/os-ecc/?S_TACT=105AGX44&S_CMP=ART

[15] https://en.wikipedia.org/wiki/C_(programming_language)

[16] https://el.wikipedia.org/wiki/C%2B%2B

[17] http://www.cplusplus.com/info/description/

[18] http://opencv.org/about.html

[19] http://docs.opencv.org/3.1.0/d1/dfb/intro.html#gsc.tab=0

[20] http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[21] http://docs.opencv.org/2.4/modules/flann/doc/flann_fast_approximate_nearest_neighbor_search.html

[22] http://docs.opencv.org/2.4/modules/highgui/doc/highgui.html?highlight=highgui

[23] http://docs.opencv.org/3.0-beta/modules/imgproc/doc/imgproc.html?highlight=imgproc

[24] http://docs.opencv.org/2.4/modules/ml/doc/ml.html

[25] http://docs.opencv.org/3.0-beta/modules/video/doc/video.html?highlight=video

[26] http://en.cppreference.com/w/cpp/language/namespace

[27] http://docs.opencv.org/master/d8/dfe/classcv_1_1VideoCapture.html#gsc.tab=0

[28] http://www.cplusplus.com/reference/cstdio/fgets/

[29] http://www.cplusplus.com/reference/cstdio/sscanf/

[30] http://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html