

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ ΚΑΙ ΥΛΙΚΟΥ



*Στερεοσκοπική όραση πραγματικού χρόνου με χρήση αναδιατασσόμενης
λογικής*

Ρεμάτσα Γεωργία

Επιβλέπων:

Καθ. Δόλλας Απόστολος

Χανιά ,2013

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	3
Περίληψη.....	3
Εισαγωγή στη στεροσκοπική όραση	3
Προηγούμενη Διπλωματική εργασία	5
Επιστημονική συνεισφορά	6
Οργάνωση της Διπλωματικής.....	7
ΚΕΦΑΛΑΙΟ 2 ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ.....	8
ΚΕΦΑΛΑΙΟ 3 ΑΝΑΛΥΣΗ ΤΗΣ ΥΠΑΡΧΟΥΣΑΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ.....	15
Γενικά.....	15
Συνάρτηση κόστους.....	15
Aggregation	16
Υπολογισμός του disparity	16
Επιλογή των τιμών των παραμέτρων	17
Διαγράμματα.....	17
Επιβεβαίωση λειτουργίας.....	19
Αξιοποίηση πόρων.....	21
ΚΕΦΑΛΑΙΟ 4 ΜΕΛΕΤΗ ΚΑΙ ΔΙΑΣΤΟΣΙΟΛΟΓΙΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	23
Το Πρόβλημα.....	23
Ελάχιστο βάθος.....	24
Πώς επηρεάζει το MAX disparity τους πόρους	27
Πώς επηρεάζει η ανάλυση τους πόρους	29
Ανάλυση των πόρων του AD-Census	31
ΚΕΦΑΛΑΙΟ 5 ΥΛΟΠΟΙΗΣΗ	35
Γενικά.....	35
Reduction Tree.....	35
Εφαρμογές του Reduction Tree	39
Στερεοσκοπικές κάμερες	39
Επιλογή του format	39
Διόρθωση εικόνας.....	40
Διαμόρφωση καμερών.....	42
Σύστημα πραγματικού χρόνου.....	42
Λειτουργία του συστήματος	43

Οργάνωση της μνήμης	44
Αποτελέσματα	45
Ανάλυση των πόρων.....	46
Ανάλυση της απόδοσης.....	50
ΚΕΦΑΛΑΙΟ 6 ΕΠΙΒΕΒΑΙΩΣΗ ΛΕΙΤΟΥΡΓΙΑΣ	51
ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	57
ΒΙΒΛΙΟΓΡΑΦΙΑ	58

Κεφάλαιο 1

Εισαγωγή

Περίληψη

Τις τελευταίες δεκαετίες, η στερεοσκοπική όραση έχει γνωρίσει μεγάλη άνθιση και χρησιμοποιείται σε μια ευρεία περιοχή εφαρμογών. Στα στερεοσκοπικά συστήματα χρησιμοποιούνται δύο κάμερες τοποθετημένες η μία δίπλα στην άλλη και με βάση τις διαφορές μεταξύ των δυο εικόνων που παράγονται, δίνεται η δυνατότητα εκτίμησης της απόστασης των αντικειμένων από την κάμερα. Στην βιβλιογραφία υπάρχει πληθώρα υλοποιήσεων 3-D στερεοσκοπικών συστημάτων με χρήση αναδιατασσόμενης λογικής, όμως στις περισσότερες περιπτώσεις απαιτείται μεγάλος αριθμός πόρων. Στην παρούσα διπλωματική εργασία γίνεται μελέτη και διαστασιολόγηση των προβλημάτων μιας προηγούμενης αρχιτεκτονικής και εξερευνούνται νέοι τρόποι σχεδίασης. Έγινε επανασχεδίαση των τμημάτων, όπου η κατανάλωση των πόρων είναι κρίσιμη, μειώνοντας τους σημαντικά. Ο βασικός αλγόριθμος για τον υπολογισμό του disparity είναι ο AD-Census Transform. Υλοποιήθηκε πραγματικό σύστημα σε χαμηλού κόστους πλατφόρμα εξοπλισμένη με μικρή FPGA, συνδεδεμένη με στερεοσκοπική κάμερα και ανάλυση 640x480. Τα δεδομένα εισόδου που παρέχονται από τις κάμερες, καθώς και το τελικό αποτέλεσμα αποθηκεύονται σε μια DDR2 SDRAM. Η επεξεργασία των δεδομένων εισόδου γίνεται σε πραγματικό χρόνο και το αποτέλεσμα προβάλλεται σε οθόνη μέσω διεπαφής HDMI.

Γενικά

Στο κεφάλαιο αυτό γίνεται μια μικρή εισαγωγή στη στερεοσκοπική όραση και στον τρόπο με τον οποίο αυτή λειτουργεί. Επιπλέον παρουσιάζεται η προηγούμενη διπλωματική εργασία, όπως εκπονήθηκε από τον κ. Σωτήρη Θωμά, της οποίας η παρούσα εργασία αποτελεί συνέχεια.

Εισαγωγή στη στερεοσκοπική όραση

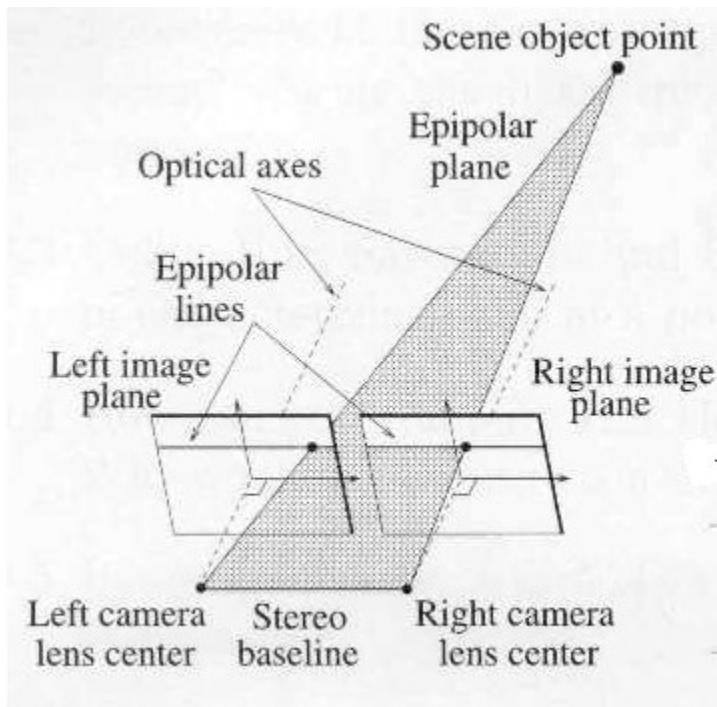
Η στερεοσκοπική όραση συνιστά ένα από τα πιο ενεργά πεδία της μηχανικής όρασης. Υπολογίζει την απόσταση σε σχέση με την κάμερα, των αντικειμένων που απεικονίζονται. Αυτό επιτυγχάνεται χρησιμοποιώντας δυο ίδιες κάμερες, οι οποίες είναι τοποθετημένες η μία δίπλα στην άλλη σε μια σχετική απόσταση και παρέχουν δυο πανομοιότυπες αλλά όχι ίδιες εικόνες με αποτέλεσμα την αντίληψη του βάθους.

Ο τρόπος αυτός ουσιαστικά προσομοιώνει την λειτουργία της ανθρώπινη όρασης, στην οποία λόγω της διαφορετικής θέσης των ματιών στο κεφάλι, δημιουργούνται δυο διαφορετικές εικόνες της σκηνής. Οι διαφορές αυτές είναι γνωστές και ως binocular disparity και δίνουν την απαιτούμενη πληροφορία στον εγκέφαλο, ώστε να υπολογίσει το βάθος.

Οποιοδήποτε σημείο μιας σκηνής ,το οποίο είναι ορατό και από τις δυο κάμερες, θα προβληθεί σε ένα ζεύγος από σημεία στις δύο εικόνες. Η μια εικόνα αποτελεί την reference και η άλλη την non-reference . Το μεγαλύτερο πρόβλημα έγκειται στο να βρούμε αυτά τα δύο σημεία . Το πρόβλημα αυτό,είναι γνωστό και σαν correspondence πρόβλημα. Η διαφορά στις θέσεις μεταξύ αυτών των σημείων στις δυο στερεοσκοπικές εικόνες αποτελεί το Disparity. Υπάρχουν πάρα πολλά εμπόδια όπως οι αντανakλάσεις, οι σκιές , οι συνθήκες φωτισμού, ο θόρυβος που κάνουν αυτή τη διαδικασία δύσκολη και ιδίως όταν οι εικόνες προέρχονται από πραγματικές κάμερες.

Όπως αναφέραμε, η στερεοσκοπική όραση χρησιμοποιεί δύο κάμερες η μια δίπλα στην άλλη οι οποίες απέχουν μεταξύ τους κατά μια απόσταση b (baseline). Οι εικόνες που προκύπτουν, προβάλλουν την ίδια σκηνή, απλώς τα αντικείμενα βρίσκονται σε διαφορετική θέση. Στόχος είναι να βρούμε όλα τα συμβατά ζεύγη που απεικονίζουν το ίδιο σημείο, ώστε με βάση το disparity να μπορούμε να εξάγουμε την πληροφορία για το βάθος μέσω τριγωνισμού (triangulation) . Το Disparity είναι άρρηκτα συνδεδεμένο με το βάθος. Συγκεκριμένα, μεγάλα disparities αντιστοιχούν σε αντικείμενα πιο κοντά στην κάμερα, ενώ μικρά disparities αντιστοιχούν σε αντικείμενα που βρίσκονται πιο μακριά από την κάμερα.

Οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί απαιτούν εικόνες οι οποίες είναι ευθυγραμμισμένες (rectified), δηλαδή ικανοποιούν τον περιορισμό της επιπολικής γεωμετρίας. Στην εικόνα 1, παρουσιάζουμε τη βασική γεωμετρία των δύο καμερών. Οι εικόνες θεωρούνται συνεπίπεδες (coplanar), δηλαδή δεν υπάρχει καμιά περιστροφή μεταξύ των καμερών .Το επίπεδο που περνάει από το κέντρο των καμερών και από το σημείο στη σκηνή, αποτελεί το επιπολικό επίπεδο(Epipolar plane). Η τομή του επιπέδου με το επίπεδο των προβολών προσδιορίζει την επιπολική γραμμή. Επομένως, η αναζήτηση των συμβατών ζευγών περιορίζεται σε μια γραμμή και ο χώρος αναζήτησης μειώνεται από 2D σε 1D.



Εικόνα 1 . Βασική Γεωμετρία των καμερών.

Στην περίπτωση που οι δύο εικόνες δεν ικανοποιούν την επιπολική γεωμετρία, τότε θα πρέπει να γίνουν διάφοροι μετασχηματισμοί, ώστε οι δύο εικόνες να έρθουν στο ίδιο επίπεδο. Η διαδικασία αυτή είναι γνωστή ως διόρθωση εικόνας (image rectification) και χρησιμοποιεί της παραμέτρους που προκύπτουν από την βαθμονόμηση των καμερών (calibration).

Προηγούμενη Διπλωματική εργασία

Στην προηγούμενη διπλωματική εργασία περιγράφεται η σχεδίαση και η υλοποίηση ενός 3D stereo vision συστήματος, το οποίο εκμεταλλεύεται αποδοτικά τους πόρους μίας FPGA. Σε επίπεδο place and route, η σχεδίαση έχει πετύχει μεγάλο ρυθμό επεξεργασίας για πολύ μεγάλη ανάλυση, ενώ ο έλεγχος και η επικύρωση του συστήματος έχει πραγματοποιηθεί για μεσαία ανάλυση.

Ο αλγόριθμος που χρησιμοποιείται, ανήκει στην κατηγορία των local αλγορίθμων και αποτελείται από τέσσερα μέρη :

Το πρώτο μέρος αναφέρεται στη συνάρτηση κόστους (matching cost), που χρησιμοποιείται για τον υπολογισμό του disparity. Η συνάρτηση αυτή είναι ένας συνδυασμός μεταξύ AD (absolute difference) και Census Transform. Μέσα από την

διαδικασία παράγονται τα αρχικά κόστη και μέσω της Winner –Take –All στρατηγικής επιλέγουμε το disparity με το μικρότερο κόστος.

Στο δεύτερο μέρος, το κόστος γίνεται aggregate .Επιπλέον, η υλοποίηση του είναι παραθυρική και στηρίζεται στο γεγονός ότι γειτονικά pixel έχουν το ίδιο disparity.

Στο τρίτο μέρος, υπολογίζεται το disparity και πραγματοποιείται Left/Right consistency check (LRC). Η μέθοδος αυτή εφαρμόζεται μετά την εφαρμογή της συνάρτησης κόστους και είναι ανεξάρτητη από αυτή. Σε αυτή την περίπτωση, παίρνουμε την non-reference εικόνα ως reference και εξετάζουμε αν το νέο disparity είναι ίδιο με το προηγούμενο.

Τέλος, στο τελευταίο μέρος υλοποιήθηκε ένας scan-line belief propagation αλγόριθμος, ο οποίος συμπληρώνει τα occluded pixel που εντοπίστηκαν στο προηγούμενο βήμα. Το τρίτο και το τέταρτο βήμα χρησιμοποιούνται για την βελτίωση της ποιότητας του τελικού αποτελέσματος.

Το σύστημα είναι πλήρως pipelined, λαμβάνει ένα ζεύγος εικονοστοιχείων σε κάθε κύκλο ρολογιού και μετά από μια αρχική καθυστέρηση παράγει ένα disparity ανά κύκλο. Το σύστημα είναι πλήρως παραμετροποιημένο. Συγκεκριμένα, δίνει τη δυνατότητα επιλογής μια μεγάλης περιοχής αναλύσεων και μπορούμε να προσαρμόσουμε πλήρως όλες τις παραμέτρους που χρειάζονται σε κάθε βήμα του αλγορίθμου. Αναφέρουμε επιπλέον, ότι δίνει τη δυνατότητα να αφαιρέσουμε ή να προσθέσουμε μέρη του αλγορίθμου. Τα δύο τελευταία μέρη, που έχουμε προαναφέρει, θα μπορούσαν για παράδειγμα να υπάρχουν ή να αφαιρεθούν από το σύστημα.

Εκτός από προσομοίωση, έχει γίνει και έλεγχος και επιβεβαίωση του συστήματος σε μια Virtex 5 XC5VLX110T, καθώς και σε μια Spartan 3 1000 FPGA. Το σύστημα έχει ελεγχθεί για 400x320 ανάλυση στην πρώτη περίπτωση και για 100x40 στη δεύτερη περίπτωση. Για τον έλεγχο του αλγορίθμου, διάφορα datasets έχουν προφορτωθεί σε μία Block RAM και μετά την επεξεργασία των δεδομένων γράφεται το αποτέλεσμα σε μια FIFO. Τα δεδομένα τις FIFO, στην συνέχεια, στέλνονται μέσω RS232 σε έναν υπολογιστή, όπου και βλέπουμε το αποτέλεσμα με τη μορφή εικόνων.

Επιστημονική συνεισφορά

Η στερεοσκοπική όραση βρίσκει εφαρμογή σε πολλές επιστημονικές περιοχές, όπως η ρομποτική, τα συστήματα πλοήγησης, όπου η αναγνώριση και ο προσδιορισμός της σχετικής θέσης των αντικειμένων έχουν ιδιαίτερη σημασία. Η εξερεύνηση του διαστήματος βασιζόμενη σε δορυφορικά συστήματα, τα αυτόνομα οχήματα, τα συστήματα ασφαλείας και παρακολούθησης αποτελούν κάποιες από τις λίγες εφαρμογές της.

Έχουν αναπτυχθεί πολλοί αλγόριθμοι, τόσο σε hardware όσο και σε software, που καλύπτουν σε μεγάλο ποσοστό τις ανάγκες που προκύπτουν, ωστόσο υπάρχουν πολύ

λίγα συστήματα το οποία ενσωματώνουν αυτούς τους αλγορίθμους σε πραγματικό χρόνο. Το κύριο εμπόδιο που αναδύεται, αφορά το μεγάλο κόστος σε πόρους στο μεν hardware ,στο δε software προκύπτει η αδυναμία να επεξεργαστούν μεγάλο αριθμό από frames ανά δευτερόλεπτο, αν και η ποιότητα της εικόνας είναι συγκριτικά πολύ καλύτερη.

Στην παρούσα διπλωματική εργασία καλύπτεται η υλοποίηση ενός πλήρως ενσωματωμένου συστήματος πραγματικού χρόνου, το οποίο υλοποιείται σε μια μόνο FPGA (Field Programmable Gate Array). Οι εικόνες παρέχονται από μια στερεοσκοπική κάμερα, η οποία συνδέεται με την fpga και επεξεργάζεται σε πραγματικό χρόνο τα video frames που παρέχονται.

Παρέχεται,επιπλέον, μια λεπτομερής ανάλυση των πόρων του συστήματος σε σχέση με το disparity της εικόνας και μία βελτιωμένη αρχιτεκτονική, η οποία χρησιμοποιεί Reduction Trees.

Οργάνωση της Διπλωματικής

Στο κεφάλαιο 2 δίνονται κάποιες τεχνικές πληροφορίες για το πρόβλημα και παρουσιάζονται οι υπάρχουσες έρευνες πάνω σε αυτό.

Στο κεφάλαιο 3 γίνεται ανάλυση της υπάρχουσας αρχιτεκτονικής, όπως αυτή έχει σχεδιαστεί και υλοποιηθεί από τον κ. Σωτήρη θωμά και γίνεται μια μικρή εισαγωγή στα προβλήματα που καθιστούν αδύνατη την εφαρμογή της στο σύστημα πραγματικού χρόνου που θα παρουσιάσουμε.

Στο κεφάλαιο 4 γίνεται μελέτη και διστασιολόγηση του προβλήματος και εξερευνούνται νέες Αρχιτεκτονικές που αντιμετωπίζουν τα προβλήματα του κεφαλαίου 3.

Στο κεφάλαιο 5 παρουσιάζεται η σχεδίαση της νέας αρχιτεκτονικής και τονίζονται οι διαφορές με την προηγούμενη.

Στο κεφάλαιο 6 παρουσιάζεται η μεθοδολογία με την οποία γίνεται επιβεβαίωση της λειτουργίας και των αποτελεσμάτων.

Τέλος στο κεφάλαιο 7 παρουσιάζονται τα συμπεράσματα και η μελλοντική δουλειά που επεκτείνουν και βελτιώνουν την υπάρχουσα δουλειά.

Κεφάλαιο 2

Σχετική έρευνα

Σε αυτό το κεφάλαιο, θα αναφέρουμε το τι έχει γίνει μέχρι σήμερα πάνω στο πρόβλημα του Stereo Vision, όσον αφορά τις fpga. Για λόγους πληρότητας γίνεται αναφορά και στην σχετική έρευνα που υπάρχει και στην προηγούμενη διπλωματική εργασία.

[1] Computer Vision Algorithms on Reconfigurable Logic Arrays.

Σε αυτή την δημοσίευση, μελετώνται παράλληλες αρχιτεκτονικές για τρεις αλγόριθμους μηχανικής όρασης, καθένας εκ των οποίων χρησιμοποιείται σε διαφορετικές εργασίες. Η πλατφόρμα που χρησιμοποιήθηκε είναι η Splash2 CCM (Custom Computing Machine) ,η οποία έχει 17 FPGAs, κάθε μια εκ των οποίων αποτελείται από PE (Processing Element).

Ο πρώτος αλγόριθμος χρησιμοποιείται για την 2 D συνέλιξη, όπου τα μερικά γινόμενα υπολογίζονται στα PEs.Ο Δεύτερος αλγόριθμος χρησιμοποιείται για image segmentation.Τέλος , ο τρίτος αλγόριθμος που εξετάζεται αφορά την ταυτοποίηση των δακτυλικών αποτυπωμάτων.

Η υλοποίηση παρουσιάζει βελτίωση κατά τέσσερις τάξης μεγέθους, σε σχέση με το SPARCstation.

[2] Low-Cost stereo Vision on an FPGA

Σε αυτή την εργασία παρουσιάζεται μια χαμηλού κόστους αρχιτεκτονική για Stereo Vision. Το σύστημα έχει υλοποιηθεί σε μια Spartan 3 fpga, δέχεται εικόνες από δυο CMOS κάμερες και παράγει το Disparity Map, εφαρμόζοντας το Census Transform για παράθυρο $W=13$ και μέγιστο disparity 20. Το σύστημα παράγει 40 fps για εικόνες 320x240, αξιοποιώντας το 57% των πόρων της FPGA.

[3] An FPGA-Base Implementation of Spatio-Temporal Object-Segmentation.

Η εργασία αυτή αναφέρεται σε Image Segmentation, χρησιμοποιώντας FPGAs. Ο αλγόριθμος έχει τρία μέρη, το πρώτο ανιχνεύει την κίνηση ,το δεύτερο αφορά το thresholding και το τρίτο ανιχνεύει τις ακμές. Το σύστημα υλοποιήθηκε σε μια Virtex 2 Pro XC2VP20 και έχει πετύχει 7.5 ms για εικόνες 1024x1024 με συχνότητα 133 MHz. Το throughput του συστήματος είναι 133 Mpixel/s, χρησιμοποιώντας το 60% των πόρων της FPGA.

[4] Small Vision Systems: Hardware and Implementation

Σε αυτή την εργασία παρουσιάζεται η υλοποίηση ενός Stereo Vision αλγορίθμου στο SVM (Small Vision Module), η οποία είναι μια DSP πλατφόρμα. Ο αλγόριθμος εφαρμόζει τον SAD σε εικόνες στις οποίες έχει εφαρμοστεί Laplacian of Gaussian φίλτρο. Ο αλγόριθμος ενσωματώνει επιπλέον Left/Right Consistency Check . Το εύρος του disparity μπορεί να είναι 16,24 και 32. Το SVM παρέχει επιπλέον αυτόματη βαθμονόμηση και παράγει 8 fps για εικόνες 160x120.

[5] FPGA Based Hardware Implementation Of Image Filter With Dynamic Reconfiguration Architecture.

Το σύστημα χρησιμοποιεί γενετικούς αλγορίθμους για να φιλτράρει και να αφαιρέσει το θόρυβο από μια εικόνα. Επιπλέον, χρησιμοποιεί coarse-timescale αναδιάταξη με σκοπό να εκμεταλλευτεί τις αργές αλλαγές στις συνθήκες. Η πλατφόρμα που χρησιμοποιήθηκε περιλαμβάνει μια 600K Spartan –IIE fpga, μια 4Mx16 μνήμη δεδομένων, μια ROM 1Mx16, μια αναλογική είσοδο με ADC,DAC για την έξοδο του video και RS232/PCI διεπαφές.

[6] A Real-Time Large Disparity Range Stereo-System Using FPGAs

Το σύστημα υλοποιεί το Local Weighted Phase Correlation (LWPC) αλγόριθμο. Χρησιμοποιεί δυο παράθυρα, το PTW (Primary Tracking Window) και το SRW (Secondary Roving Window). Το πρώτο χρησιμοποιείται για την εκτίμηση της θέσης ενός pixel στην εικόνα και το άλλο για να εντοπίσει μεγάλες αλλαγές στο disparity μεταξύ των εικόνων. Το εύρος του disparity μπορεί να φτάσει έως 128 pixels. Το σύστημα υλοποιήθηκε στο Transmogripher-4, το οποίο περιέχει 4 Altera Stratix S80 FPGAs. Το σύστημα παρέχει 30 fps για ανάλυση 640x480.

[7] High performance FPGA base Optical Flow calculation using the Census Transformation.

Σε αυτή την δημοσίευση παρουσιάζεται μια διαφορετική υλοποίηση του Census-Transform για Optical Flow video. Γενικά, ο αλγόριθμος εντοπίζει συμβατά pixel σε δυο διαδοχικές εικόνες και δημιουργεί ένα διάνυσμα από το σημείο που ξεκινάει μέχρι το σημείο που τελειώνει. Η δημιουργία του διανύσματος στηρίζεται σε διάφορα signatures, που δημιουργούνται για κάθε pixel από την διαφορετική εκδοχή του Census-Transform.

Υλοποιήθηκαν δυο εκδοχές του αλγορίθμου ,μία σε software και μία σε Hardware. Η κύρια διαφορά τους είναι ότι στο software η αναζήτηση των συμβατών pixel γίνεται σε

όλη την εικόνα, ενώ στο hardware γίνεται στα γειτονικά pixel. Για το software χρησιμοποιήθηκε Core 2 Duo 1.86 GHz επεξεργαστής, ενώ το Hardware υλοποιήθηκε σε μια XC2VP30 fpga της Xilinx με δυο PowerPC επεξεργαστές. Το hardware χρειάζεται 22.17 ms για το κάθε σετ και έχει πετύχει 1.8 speedup σε σχέση με το software.

[8] FPGA Design and Implementation of a Real-Time Stereo Vision System.

Σε αυτή την δημοσίευση παρουσιάζεται ένα πλήρες σύστημα, υλοποιημένο σε μια fpga. Σε αυτό έχει υλοποιηθεί rectification της εικόνας, χρησιμοποιείται το Census Transform ως βασικός αλγόριθμος για να βρεθούν τα συμβατά pixels, και για την βελτίωση της ποιότητας της εικόνας χρησιμοποιείται uniqueness check, left/right check, εκτίμηση των sub-pixel και spike removal.

Το σύστημα έχει υλοποιηθεί σε μια Xilinx Virtex-4 XC4VLX200-10 fpga. Η θεωρητική μέγιστη επίδοση είναι 230 fps με 17.24% μέσο σφάλμα των pixels, για 640x480 εικόνες.

[10] Towards Hardware Stereoscopic 3D Reconstruction, A Real-Time FPGA Computation of Disparity Map.

Στην συγκεκριμένη δημοσίευση παρουσιάζεται η υλοποίηση ενός stereo vision αλγόριθμου. Ο αλγόριθμος αποτελείται από δυο βήματα. Το πρώτο αφορά το να βρεθούν τα συμβατά pixels, όπου και χρησιμοποιείται το SAD και το δεύτερο αφορά τον υπολογισμό του 3D map και τον υπολογισμό της πραγματικής απόστασης από την κάμερα. Η εφαρμογή του SAD γίνεται σε εικόνες, στις οποίες έχει εφαρμοστεί Sobel φίλτρο για την ανίχνευση των ακμών. Το σύστημα έχει υλοποιηθεί σε μια Xilinx Virtex2 Pro XC2VP30. Για disparity 31, ανάλυση 320x240 και παράθυρο 9x9 το σύστημα πετυχαίνει 17,1 fps χωρίς την ανίχνευση ακμών και 37 fps με την ανίχνευση ακμών.

[17] A real-time stereo vision system using a tree-structured dynamic programming on FPGA.

Όπως αναφέρεται και στον τίτλο, σε αυτή την δημοσίευση παρουσιάζεται η υλοποίηση ενός tree-structured αλγορίθμου δυναμικού προγραμματισμού σε FPGAs. Ανήκει στην κατηγορία των global αλγορίθμων και έχει το πλεονέκτημα ότι μπορεί να συναγωνιστεί το ποσοστό σφάλματος των αλγορίθμων λογισμικού.

Ο αλγόριθμος χρησιμοποιεί SAD για την τοπική αντιστοίχιση των pixels και λαμβάνει υπόψη του τη σύνδεση των pixels στο δέντρο για τον υπολογισμό του disparity. Πραγματοποιεί αυτή τη διαδικασία, έχοντας και την αριστερή εικόνα ως βάση αλλά και την δεξιά. Τα σημεία που ταιριάζουν στις δύο αναζητήσεις, αποτελούν disparity map.

Μόνο τα pixels που είναι όμοια με το $R(x,y)$ χρησιμοποιούνται για τον υπολογισμό του SAD. Για τον υπολογισμό του disparity, προστίθεται στα SADs ένα επιπλέον κόστος, λόγω του smoothness, λαμβάνοντας υπόψη τη σχέση μεταξύ των pixels στο δέντρο.

Το σύστημα έχει υλοποιηθεί σε μια Xilinx XC4VLX160 και RC2000-4 από Agility για $X < 512$ και $D = 60$ και συχνότητα 216.233 MHz. Το σύστημα παρέχει 128 fps για ανάλυση 320x240 και 32.0 fps για ανάλυση 640x480.

[18] A FAST AND HIGH QUALITY STEREO MATCHING ALGORITHM ON FPGA

Σε αυτή την δημοσίευση παρουσιάζεται η υλοποίηση ενός συστήματος, το οποίο έχει υψηλό ρυθμό επεξεργασίας και χαμηλό σφάλμα. Το σύστημα έχει υλοποιηθεί σε Xilinx Virtex-6 FPGAs και για την υλοποίηση του έχουν χρησιμοποιηθεί δύο αλγόριθμοι το Cost Aggregation (CA) και το Fast locally consistent dense stereo (FLC).

Ως συνάρτηση κόστους του συστήματος, χρησιμοποιείται το Mini-Census Transform. Υπολογίζει δυο Disparity maps, ένα για την αριστερή εικόνα και ένα για την δεξιά, με βάση τα οποία επιλέγονται τα ground control points (GCPs) που χρησιμοποιούνται στον FLC αλγόριθμο.

Το σύστημα έχει υλοποιηθεί για $X < 1024$ και $D = 60$ σε μια Xilinx XC6VLX240T και η συχνότητα λειτουργίας είναι 318.259 MHz. Καταλαμβάνει το 81% των LUTs και το 47.6% από το 36Kb block RAMs. Το σύστημα επεξεργάζεται 1981.1 fps για ανάλυση 320x240 και 507.4 fps για ανάλυση 640x480 και 199.7 fps για ανάλυση 1024x768.

[19] Edge-Directed Hardware Architecture for Real-Time Disparity Map Computation.

Σε αυτή τη δημοσίευση παρουσιάζεται ο υπολογισμός ενός disparity map, το οποίο ενσωματώνει τεχνικές SAD και συνδυάζεται με τις ακμές που προκύπτουν από έναν edge detector και η πρόσβαση στην εξωτερική μνήμη, στην οποία βρίσκονται οι εικόνες, γίνεται κατά τον βέλτιστο τρόπο. Θεωρεί ότι έχει γίνει είδη rectification στις εικόνες, ενώ χρησιμοποιώντας το περίγραμμα των εικόνων μειώνεται ο χρόνος αναζήτησης και οι πόροι. Οι grayscale εικόνες μετατρέπονται σε ένα ζεύγος από δυαδικές εικόνες (ασπρόμαυρες), οι οποίες περιέχουν το περίγραμμα των αρχικών εικόνων και με βάση το οποίο υπολογίζεται το τελικό disparity map. Το σύστημα υλοποιήθηκε σε μια Xilinx ML505 FPGA Evaluation Platform πετυχαίνοντας 40 fps για εικόνες 1280x1024.

[20]Towards accurate hardware stereo correspondence: A real-time FPGA implementation of a segmentation-based adaptive support weight algorithm.

Στην συγκεκριμένη δημοσίευση, παρουσιάζεται η υλοποίηση ενός αλγορίθμου κατάλληλο για embedded vision συστήματα. Βασίζεται στον adaptive support weight (ADSW) αλγόριθμο. Αξιοποιεί τις πληροφορίες που προκύπτουν από την τμηματοποίηση της εικόνας για να υπολογίσει τα βάρη (weight), ώστε να πετύχει καλύτερη ακρίβεια στην ποιότητα της εικόνας. Χρησιμοποιεί επιπλέον το περίγραμμα για να μειώσει το χρόνο αναζήτησης και τους πόρους.

Recourses	Disparity	Resolution	Performance (fps)	Frequency
26/40 BRAMs 57% logic recourses	40 (theoretical 150)	320x240	26 MHz	
-	16,24,32	320x240	12	33 MHz
Registers:83026 LUTs(4-input):66644	128	640x480	30	
LUTs: 34 % FlipFlops : 30%	64	640x480	230	93.09.07 MHz
LUTs: 61.1 % FlipFlops : 79.9%	32	320x240	75	-

Ref	Technology
[2]	Xilinx Spartan 3
[4]	DSP: ADSP 2181
[6]	4-FPGAs Platform: BRAMS : 197/640 Transmogrifier-4 4xAltera Stratix S80 FPGAS
[8]	FPGA: Xilinx Virtex-4 XC4VLX200-10
[10]	XC2 VP30 Pro Xilinx Virtex 2

60	640x480	MHz	216.233
60	640x480 1024x768	507.4	318.259
120	1280x1024	50	100
64	640x480	30	155
64	640x480	650	201 MHz
54	640x480	227	69.788 MHz

(περιορίζεται
λόγο των
καμερών)
θεωρητικά

□	Agility	Xilinx XC4VLX160 on RC2000-4 from		LUTs: 72% BRAMs (46%)
□		Xilinx XC6VLX240T		LUTs: 81% BRAMs (47.6%)
□		Xilinx ML505 FPGA Evaluation Platform	BRAMs (20%)	LUTs: 83.8 % Slice Registers :60%
□		FPGA	BRAMs 20,3%	LUTs: 80.43 % Registers :90.01%
υλοποίηση	Προηγούμενη	Xilinx Virtex5 XC5VLX110T-1	LUTs:54 %	FlipFlops:60 % Slices: 82%
Δική μας υλοποίηση	Atlys Spartan 6 FPGA		BRAMs:74%	LUTs :91% Registers: 48%

Κεφάλαιο 3

Ανάλυση της Υπάρχουσας Αρχιτεκτονικής

Γενικά

Ο αλγόριθμος που χρησιμοποιείται ανήκει, όπως έχουμε αναφέρει, στην κατηγορία των local αλγορίθμων, δηλαδή επιλέγει τα disparities για κάθε pixel, ανεξάρτητα από τα υπόλοιπα. Θεωρεί επιπλέον ότι έχει προηγηθεί rectification στις δυο εικόνες, επομένως ο χώρος αναζήτησης μειώνεται σε μία διάσταση. Ο χώρος αυτός περιγράφεται από την παράμετρο που ορίζει το μέγιστο Disparity D_{max} .

Ο αλγόριθμος προσφέρει παραλληλισμό σε πολλά επίπεδα παρέχοντας την δυνατότητα να υπολογιστούν ταυτόχρονα οποιαδήποτε κόστη, οποιασδήποτε ανάλυσης, εφόσον δεν ξεπερνούν τον αριθμό των πόρων που παρέχονται από το εκάστοτε σύστημα. Δεδομένου ότι οι περισσότερες κάμερες παρέχουν τα pixels σειριακά, τα δεδομένα επεξεργάζονται σε streaming μορφή και παρέχει D_{max} εκτιμήσεις για το disparity σε κάθε κύκλο ρολογιού.

Στη Γενική περίπτωση, όλοι οι αλγόριθμοι που αναφέρονται στη στερεοσκοπική όραση υλοποιούν ένα υποσύνολο των παρακάτω βημάτων.

- Υπολογισμός της συνάρτησης κόστους (Matching cost)
- Aggregation του κόστους
- Υπολογισμός του disparity
- Βελτίωση των αποτελεσμάτων

Συνάρτηση κόστους

Η συνάρτηση κόστους αποτελεί ένα μέτρο ομοιότητας των pixel. Για την υλοποίηση της έχει επιλεγεί το Census Transform, σε συνδυασμό με το Absolute Difference (AD). Το AD περιγράφεται από την σχέση 3.1 και αποτελεί ένα μέτρο σύγκρισης μεταξύ των εντάσεων των δύο pixel.

$$AD = |I_r(x, y) - I_l(x + d, y)|$$

3.1

Το Census Transform [13] βασίζεται μόνο σε συγκρίσεις μεταξύ των γειτονικών pixel ενός παραθύρου, για να εξάγει τις πληροφορίες που χρειάζονται. Περιέχει δυο λογικά βήματα, το πρώτο αφορά τη δημιουργία ενός bit-string και το δεύτερο χρησιμοποιεί τα bit-strings για να βρει το καλύτερο κόστος σε μια περιοχή στη non-reference εικόνα.

Για κάθε pixel, στην reference εικόνα, υπολογίζουμε το census bit-string για το συγκεκριμένο pixel, το οποίο αποτελεί το κεντρικό bit-string. Το Census bit-string είναι ουσιαστικά ένα διάνυσμα μεγέθους W^2-1 , όπου το κάθε bit στο διάνυσμα δηλώνει αν η ένταση του pixel στη συγκεκριμένη θέση είναι μεγαλύτερη/ίση ή μικρότερη από αυτή του κεντρικού pixel. Υπολογίζουμε με τον ίδιο τρόπο τα υποψήφια bit-strings σε μια περιοχή της non-reference εικόνας. Η περιοχή αυτή είναι ίση με το μέγιστο disparity Το hamming distance μεταξύ του κεντρικού bit-string και των D_{max} bit-strings αποτελεί το κόστος.

Aggregation

Κατά την εφαρμογή του AD-Census Transform, δημιουργείται το DSI (Disparity Search Image), ο οποίος είναι ένας τρισδιάστατος χώρος μεγέθους $W \times H \times D_{max}$ που περιέχει τα κόστη όλων των D_{max} υποψηφίων για κάθε pixel (x,y) σε μια $W \times H$ εικόνα.

Αν υποθέσουμε ότι τα γειτονικά pixel έχουν το ίδιο disparity, τότε τα κόστη στην περιοχή ενός παραθύρου μπορούν να γίνουν aggregated. Επομένως, το aggregation ισοδυναμεί με το να εφαρμόσουμε ένα απλό φίλτρο στη (x,y) διάσταση του DS.

Υπολογισμός του disparity

Αφού τα διάφορα κόστη γίνουν aggregated, επιλέγουμε ως disparity για το σημείο (x,y) τη θέση d του DSI. Η επιλογή αυτή γίνεται ακολουθώντας τη Winner-Take-All στρατηγική, όπου επιλέγεται εκείνο με το μικρότερο κόστος.

Κατά γενική περίπτωση, υπάρχουν σημεία στο χώρο, όπου μπορεί να εμφανίζονται σε μια από τις δυο εικόνες, με αποτέλεσμα την ύπαρξη occluded pixels. Αυτά οδηγούν συνήθως σε λανθασμένες τιμές του disparity. Για το λόγο αυτό εφαρμόζεται Left/Right Consistency Check (LRC). Το LRC ελέγχει αν τα disparities που υπολογίστηκαν στο προηγούμενο βήμα ταιριάζουν με τα disparities που προκύπτουν, αν χρησιμοποιηθεί η άλλη εικόνα ως reference. Ουσιαστικά, η παρακάτω συνθήκη θα πρέπει να είναι αληθής για κάποια τιμή του LRC Threshold

$$|Disp_{RL}(x, y) - Disp_{LR}(Disp_{RL}(x, y), y)| < LRCthreshold$$

Η διαδικασία αυτή, ουσιαστικά αποτρέπει τις λανθασμένες τιμές για το disparity που οφείλονται σε occlusions .

Βελτίωση των αποτελεσμάτων

Οι λανθασμένες τιμές των disparities, που εντοπίστηκαν από το LRC, αντικαθίστανται από τα πιο σίγουρα disparities (confident disparities) που υπάρχουν σε μια γραμμή της εικόνας. Αυτό πραγματοποιείται με έναν απλο Scanline Belief Propagation αλγόριθμο. Ο αλγόριθμος έχει δυο ουρές, η μια περιέχει τη θέση και την κατάσταση των pixel (neighborhood Queue) και η άλλη περιέχει τα confident disparities (confident Disparities Queue). Όταν η δεύτερη ουρά γεμίσει, τότε τα confident disparities διαδίδονται στα σημεία που παρουσιάζονται ασυνέπειες.

Επιλογή των τιμών των παραμέτρων

Τα βασικά μέρη του αλγορίθμου έχουν πέντε παραμέτρους, που χρειάζονται να ρυθμιστούν. Το Dmax , το Census Window (W) και το aggregation window (Wa). Μετά από προσεκτική ανάλυση, επιλέγεται η τιμή W=9 για το Census παράθυρο και η τιμή Wa =5 για το aggregation.

Διαγράμματα

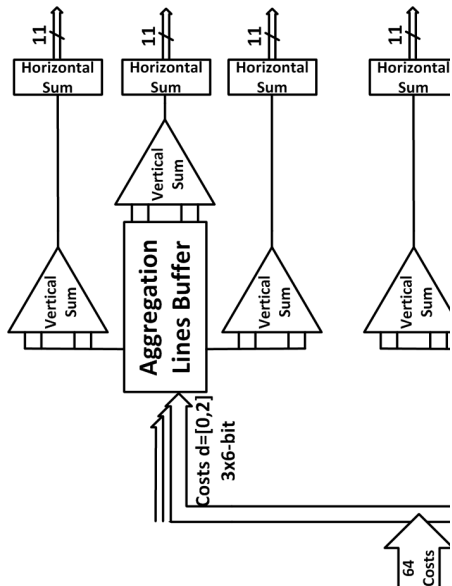
Στην εικόνα 3.1 απεικονίζεται το γενικό διάγραμμα του datapath που αφορά το βασικό στάδιο του αλγορίθμου.

Το σύστημα δέχεται ως είσοδο δυο 8-bit τιμές, που αντιστοιχούν στα pixel από τις δύο εικόνες. Το Lines Buffer αποθηκεύει W-1 γραμμές τις εικόνες, το καθένα σε μια BRAM, μετατρέποντας το με αυτό τον τρόπο σε ένα διάνυσμα στήλης μεγέθους W. Το Window Buffer μετατρέπει τις στήλες αυτές σε ένα W^2 πίνακα. Ο πίνακας αυτός οδηγείται στη μονάδα Census Bitstring Generator, όπου και υπολογίζεται το Census bit string.

Η μονάδα Central pixels /Bistrings FIFO αποθηκεύει Dmax non-reference census bitstrings και το κεντρικό εικονοστοιχείο του παραθύρου. Τα Dmax bitstring και τα κεντρικά εικονοστοιχεία, τόσο της reference όσο και της non-reference εικόνας, οδηγούνται στη μονάδα Compute Cost. Η μονάδα αυτή είναι υπεύθυνη για το XOR/summing που απαιτείται για τον υπολογισμό του Hamming Distance, καθώς και τον υπολογισμό του AD. Τα δύο αυτά κομμάτια προστίθενται και κανονικοποιούνται .

Στη μονάδα του aggregation παράγονται Dmax aggregate κόστη. Συγκεκριμένα, χρησιμοποιούνται 22 Lines Buffers για να αποθηκεύσουν τα κόστη που παράγονται από το προηγούμενο βήμα. Κάθε Line Buffer αντιστοιχίζεται σε τρία κόστη. Ουσιαστικά

μετατρέπουν τη ροή δεδομένων σε διανύσματα στήλης μεγέθους W_a . Κάθε στήλη, στη συνέχεια, προστίθεται ξεχωριστά στη μονάδα Vertical Sum και οδηγούνται έπειτα στη μονάδα Horizontal Scan, όπου καθυστερούνται και προστίθενται με την W_a ροή δεδομένων που ακολουθεί.



[Type a quote from the document or the summary of an interesting point. You can position the text box anywhere in the document. Use the Text Box Tools tab to change the formatting of the pull quote

Στην εικόνα 3.2 (a) παρουσιάζουμε την μονάδα LRC. Τα disparity της non-reference εικόνας προκύπτουν από τα left-to-right κόστη, τα οποία παράγονται από τη μονάδα LRC Buffer. Η μονάδα WTA χρησιμοποιεί comparator trees για να επιλέξει το μικρότερο κόστος. Το Reference disparity καθυστερείται μέχρι τα non-reference disparity να γεμίσουν τον NonReference Disparities Buffer. Η

απόλυτη διαφορά μεταξύ του $Disp_{RL}(x,y)$ και $Disp_{LR}(x,y)$, που είναι μικρότερη από ένα συγκεκριμένο κατώφλι, μας δίνει τα false matches.

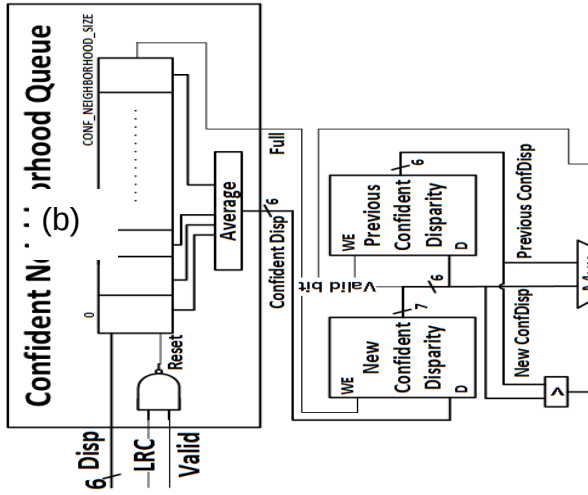
Ο αλγόριθμος για το Scanline Belief Propagation υλοποιήθηκε, όπως παρουσιάζεται στην εικόνα 3.2(b). Η λειτουργία αυτής της μονάδας στηρίζεται σε δύο ουρές την Confident Neighborhood Queue και την Neighborhood Queue. Στην πρώτη κρατούνται τα disparities, που γίνονται αποδεκτά από το LRC. Σε κάθε κύκλο υπολογίζεται ο μέσος όρος των Confident disparities και στη συνέχεια διαδίδεται στα unconfident disparities της neighborhood queue.

Επιβεβαίωση Λειτουργίας

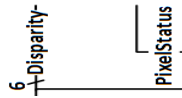
Για την επιβεβαίωση της συγκεκριμένης υλοποίησης ακολουθήθηκαν δύο τρόποι, το ένα έγινε σε επίπεδο προσομοίωσης, όπου τα διάφορα datasets διαβάζονται μέσω testbench και αντίστοιχα μέσω testbench παράγεται και το αποτέλεσμα, που αποτελεί μια νέα εικόνα.

Στον δεύτερο τρόπο, οι Block RAMs της FPGA αρχικοποιήθηκαν με κάποιο συγκεκριμένο dataset κάθε φορά. Ο αλγόριθμος επεξεργάζεται τα δεδομένα και το

αποτελέσματα γράφονται σε μια FIFO, όπου και στη συνέχεια στέλνονταν μέσω RS232 σε κάποιον υπολογιστή. Επειδή το bitrate της RS232 είναι χαμηλό, η FIFO έπρεπε να είναι αρκετά μεγάλη για να χωράει ολόκληρο το αποτέλεσμα.



Εικό
να
3.2



(a)

Αξιοποίηση πόρων

Στον πίνακα 3.1, παρουσιάζεται αναλυτικά πως αξιοποιούνται οι πόροι σε μια Virtex XC5VLX110T συνολικά και για το κάθε κομμάτι του αλγορίθμου ξεχωριστά, ενώ στον πίνακα 3.2 παρουσιάζεται αντίστοιχα, πως αξιοποιούνται οι πόροι σε μια Spartan 6 XC6SLX45 fpga για τα ίδια κομμάτια. Το ποσοστό που αναγράφεται προκύπτει μεταξύ του κάθε μέρους του αλγορίθμου και των διαθέσιμων πόρων. Όπως φαίνεται, δεν γίνεται αναφορά στη συνολική κατανάλωση των πόρων, καθώς ο αριθμός των πόρων της συγκεκριμένης fpga είναι πολύ μικρός και αυτό καθιστά το post place and route στη σχεδίαση αδύνατο. Ωστόσο δίνουμε μια μικρή εκτίμηση των πόρων αυτών στον πίνακα 3.3, όπως αυτά προκύπτουν από το Synthesize των εργαλείων της Xilinx .

Dmax=64, Wc =9 ,Wa=5	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	69,120	69,120	148
Total consumed	37,986 out of 69,120 (55%)	41,792 out of 69,120 (60%)	59 out of 148 (40%)
AD Census	25,135 out of 37,986(66%)	29,167 out of 41,792(70%)	8 out of 59(14%)
Aggregation	6,547 out of 37,986 (17%)	7,312 out of 41,792 (17%)	51 out of 59(86%)
Left/Right Check	4,638 out of 37,986 (12%)	4,734 out of 41,792 (11%)	0 out of 59 (0%)
Scanline Belief Propagation	543 out of 37,986 (1.5%)	634 out of 41,792 (1.5%)	0 out of 59 (0%)

Πίνακας 3.1. Αξιοποίηση πόρων στη Virtex XC5VLX110T για κάθε υποσύστημα

Dmax=64, Wc =9 ,Wa=5	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
AD Census	22.943 (84 %)	30.219 (55%)	8 (6 %)
Aggregation	8.303	6.449	66
Left/Right Check	4.345	3.459	0
Scanline Belief Propagation	443	449	0

Πίνακας 3.2 . Αξιοποίηση πόρων στην Spartan 6 XC6SLX45 για κάθε υποσύστημα

Dmax=64, Wc =9 ,Wa=5	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total consumed	36.632	40317	74
AD Census	22.943	30.219	8
Aggregation	8.646	6.218	66
Left/Right Check	4864	3459	0
Scanline Belief Propagation	445	448	0

Το AD-Census καταλαμβάνει το μεγαλύτερο μέρος των πόρων που απαιτεί ο αλγόριθμος στο σύνολο, γεγονός που το καθιστά και το πιο ακριβό κομμάτι.

Κεφάλαιο 4

Μελέτη και διαστασιολόγηση του προβλήματος

Το Πρόβλημα

Έχοντας στη διάθεση μας την Spartan 6 για την υλοποίηση του συστήματος προκύπτει ένα μεγάλο πρόβλημα, η διαθεσιμότητα των πόρων, και συγκεκριμένα των LUTs. Παρουσιάζουμε στον πίνακα 4.1 όλους τους διαθέσιμους πόρους των δύο FPGA. Και στις δύο περιπτώσεις, κάθε slice περιέχει 4 LUTs. Το κάθε LUT είναι 6 εισόδων, γεγονός που δεν επηρεάζει την αξιοποίησή τους στην Spartan 6. Βέβαια όπως φαίνεται και από τους πίνακες, η Virtex 5 διαθέτει πολλά περισσότερα slices, άρα και LUTs. Συγκεκριμένα η Virtex 5 περιέχει περίπου 60% περισσότερα LUTs. Επιπλέον, έχοντας μια εκτίμηση των LUTs που απαιτούνται στην Spartan 6, όπως αυτή προκύπτει από τον πίνακα 3.3 του προηγούμενου κεφαλαίου, καταλήγουμε στο συμπέρασμα ότι η υπάρχουσα αρχιτεκτονική είναι αδύνατο να υλοποιηθεί στη συγκεκριμένη συσκευή.

	Slices	Registers	LUTs	BRAMs
XC5VLX110T	17.280	69.120	69.120	148
Atlys	6.822	54.576	27.288	116

Πίνακας 4.1 Διαθέσιμοι πόροι στις δύο FPGA

Όπως έχουμε ήδη αναφέρει, στην υπάρχουσα υλοποίηση το σύστημα δέχεται μόνο δυο pixels στην είσοδο κάθε φορά, τα οποία αντιστοιχούν στις δύο εικόνες και επεξεργάζεται τα δεδομένα σε streaming μορφή. Επειδή στη παρούσα διπλωματική πραγματευόμαστε ένα ενσωματωμένο σύστημα πραγματικού χρόνου, απαιτείται η αποθήκευση των δύο εικόνων καθώς και του αποτελέσματος. Αν χρησιμοποιήσουμε BRAMs για αυτή την αποθήκευση, τότε αυτό θα περιόριζε το μέγεθος τις εικόνες σε πολύ μικρές εικόνες καθώς ο αριθμός των BRAMs δεν επαρκεί.

Ελάχιστο Βάθος

Η επιλογή του μέγιστου disparity εξαρτάται κυρίως από την ανάλυση που έχει η εικόνα και συγκεκριμένα από το frame width .Επιπλέον, το μέγιστο disparity παίζει πολύ μεγάλο ρόλο στο εύρος του βάθους που μπορούμε να υπολογίσουμε. Οι σχέσεις (1) και (2) μας δίνουν το μέγιστο και το ελάχιστο βάθος που μπορεί να εντοπίσει η κάμερα.

$$Depth_{\min} = \frac{b * f}{D_{\max} * DP_{hor}} \quad (1)$$

$$Depth_{\max} = \frac{b * f}{D_{\min} * DP_{hor}} \quad (2)$$

$$DP_{hor} = \frac{\text{Active pixel array area}}{\text{Image width}} \quad (3)$$

Το DP_{hor} είναι το pixel horizontal dot pitch, δηλαδή η οριζόντια απόσταση μεταξύ δύο γειτονικών pixel, το οποίο εξαρτάται και από την ανάλυση και μπορούμε να το υπολογίσουμε σύμφωνα με τη σχέση (3). D_{\max} είναι το μέγιστο disparity, b είναι το baseline της κάμερας, δηλαδή η απόσταση από το κέντρο των δύο καμερών και f το focal length. Στον πίνακα 4.3 παρουσιάζουμε όλες τις σταθερές παραμέτρους των καμερών που χρησιμοποιούνται.

Parameter	Value
Baseline b	63 mm
Focal length f	3,79 mm

Active pixel array area 3,56 mm

Πίνακας 4.3 Παράμετροι των Aptina MT9D112

Για κάθε σημείο y , μπορούμε να υπολογίσουμε το βάθος, χρησιμοποιώντας την εξίσωση (1) όπου βέβαια χρησιμοποιούμε την τιμή του disparity που υπολογίζεται.

$$Depth = \frac{b * f}{d * DP_{hor}}$$

Η επιλογή του μέγιστου disparity δεν μπορεί να γίνει αυθαίρετα. Θα πρέπει να λάβουμε υπόψη τις απαιτήσεις που υπάρχουν κάθε φορά. Συγκεκριμένα, το ελάχιστο βάθος που μπορούν να εντοπίσουν οι κάμερες εξαρτάται σε μεγάλο βαθμό από το disparity, όπως προείπαμε. Για μια εικόνα 640x480 και για το μέγιστο disparity να παίρνει τιμές από 10 έως 64, τότε το ελάχιστο βάθος θα κυμαίνεται από 4,29 m μέχρι 0.67 m, αντίστοιχα. Ενώ, το μέγιστο βάθος θα είναι σε κάθε περίπτωση ίσο με 42.9 m.

Στο διάγραμμα 4.1 παρουσιάζουμε τη σχέση μεταξύ ελάχιστου βάθους και των μέγιστων disparities. Επομένως, η επιλογή μικρής τιμής για το max disparity, θα έχει ως αποτέλεσμα να μην μπορούμε να εντοπίσουμε αντικείμενα σε πολύ κοντινή απόσταση. Η επιλογή του μέγιστου disparity εξαρτάται και σε ένα βαθμό και από την ανάλυση της εικόνας.

Στο διάγραμμα 4.2 βλέπουμε πώς μεταβάλλεται το ελάχιστο βάθος για διάφορα frame widths και διάφορα disparities. Όσο μικραίνει το μέγιστο disparity, μπορούμε να εντοπίσουμε μικρότερες αποστάσεις για κάθε ανάλυση. Όπως βλέπουμε, για μια εικόνα 640x480 και disparity 10 το ελάχιστο βάθος ξεπερνάει τα 4 μέτρα.

Το ελάχιστο βάθος που μπορεί να εντοπίσουν η κάμερες μπορεί να διατηρηθεί μεγαλώνοντας το μέγιστο disparity, κάτι τέτοιο όμως έχει μεγάλες επιπτώσεις σε πόρους, όπως θα δούμε παρακάτω. Στο να διατηρήσουμε το ίδιο ελάχιστο βάθος, θα μπορούσε να συμβάλει και η μείωση του baseline, κάτι τέτοιο όμως είναι αδύνατο να εφαρμοστεί, διότι στην περίπτωσή μας είναι συγκεκριμένο.

Διάγραμμα 4.1 Σχέση μεταξύ του ελάχιστου βάθους και του μέγιστου disparity

Διάγραμμα 4.2 Σχέση μεταξύ frame width και του μέγιστου disparity

Πώς επηρεάζει το MAX Disparity τους πόρους

Όλες οι μονάδες που υλοποιούν τα διάφορα μέρη του αλγορίθμου εξαρτώνται άμεσα από το disparity. Στον AD- Census Transform χρειάζεται να υπολογιστούν τα Dmax κόστη, παράλληλα. Αυτό απαιτεί Dmax μονάδες που υπολογίζουν το Hamming distance και το Absolute-Difference (AD). Το κομμάτι του αλγορίθμου που υλοποιεί το aggregation απαιτεί να γίνουν aggregate τα Dmax κόστη, που υπολογίστηκαν στο προηγούμενο βήμα, όπου στη συνέχεια στο τρίτο μέρος, συγκρίνονται και επιλέγεται εκείνο με το μικρότερο κόστος. Στις εικόνες 4.3, 4.4 και 4.5 φαίνεται πώς μεταβάλλεται το κάθε κομμάτι του αλγορίθμου, για διάφορες τιμές του μέγιστου Disparity.

Οι τιμές αυτές προκύπτουν μετά από synthesize των εργαλείων της Xilinx για την Spartan 6, καθώς το place and route είναι αδύνατο, λόγω των περιορισμών που έχουμε στους πόρους της συγκεκριμένης fpga.

Από τα διαγράμματα, είναι εμφανές ότι με την αύξηση του disparity αυξάνονται και οι απαιτούμενοι πόροι (Registers και LUTs) με μια σχεδόν γραμμική σχέση για τα τρία πρώτα μέρη του αλγορίθμου. Δεν συμβαίνει το ίδιο για Belief Propagation, καθώς οι πόροι σε αυτή την περίπτωση μεταβάλλονται ελάχιστα. Βέβαια, το Belief Propagation, όπως είδαμε στο κεφάλαιο 3, καταλαμβάνει ένα πολύ μικρό μέρος των συνολικών πόρων. Το μέγιστο disparity επηρεάζει τις BRAMs μόνο στο κομμάτι aggregation, ενώ στο Census οι BRAMs διατηρούνται σταθερές και ανεξάρτητες από το μέγιστο Disparity. Τα υπόλοιπα μέρη του αλγορίθμου δεν καταναλώνουν BRAMs και επομένως δεν παρουσιάζονται.

Εικόνα 4.3 . Σχέση μεταξύ του MAX Disparity και LUTs για το κάθε κομμάτι του αλγορίθμου.

Εικόνα 4.4 . Σχέση μεταξύ του MAX Disparity και Registers για το κάθε κομμάτι του αλγορίθμου.

Εικόνα 4.5 . Σχέση μεταξύ του MAX Disparity και BRAMs για τα δυο κομμάτια του αλγορίθμου.

Πώς επηρεάζει η ανάλυση τους πόρους:

Στις εικόνες 4.6, 4.7, 4.8 φαίνεται πώς μεταβάλλονται οι πόροι για το κάθε κομμάτι του αλγορίθμου για διάφορες τιμές του frame width. Το disparity διατηρήθηκε σταθερό σε όλες τις περιπτώσεις και ίσο με 64.

Το frame width δεν επηρεάζει σε μεγάλο βαθμό τους πόρους (Registers και LUTs). Οι μεταβολές για αυτήν την περίπτωση, για όλα τα κομμάτια του αλγορίθμου είναι ελάχιστες. Επηρεάζει, όμως, τον αριθμό των BRAMs και συγκεκριμένα στο κομμάτι του aggregation. Με την αύξηση της ανάλυσης, παρατηρούμε ότι αυξάνεται ο αριθμός αυτός. Στην περίπτωση του Census ο αριθμός των BRAMs διατηρείται σταθερός μέχρι και για frame width 640. Για frame width 1280, ο αριθμός αυτός διπλασιάζεται. Στην τελευταία περίπτωση, τα υπόλοιπα μέρη του αλγορίθμου δεν καταναλώνουν BRAMs και επομένως δεν παρουσιάζονται.

Εικόνα 4.6 . Σχέση μεταξύ Frame Width και LUTs για το κάθε κομμάτι του αλγορίθμου.

Εικόνα 4.7 . Σχέση μεταξύ frame Width και Registers για το κάθε κομμάτι του αλγορίθμου.

Εικόνα 4.8 . Σχέση μεταξύ Frame Width και BRAMs για τα δυο κομμάτια του αλγορίθμου.

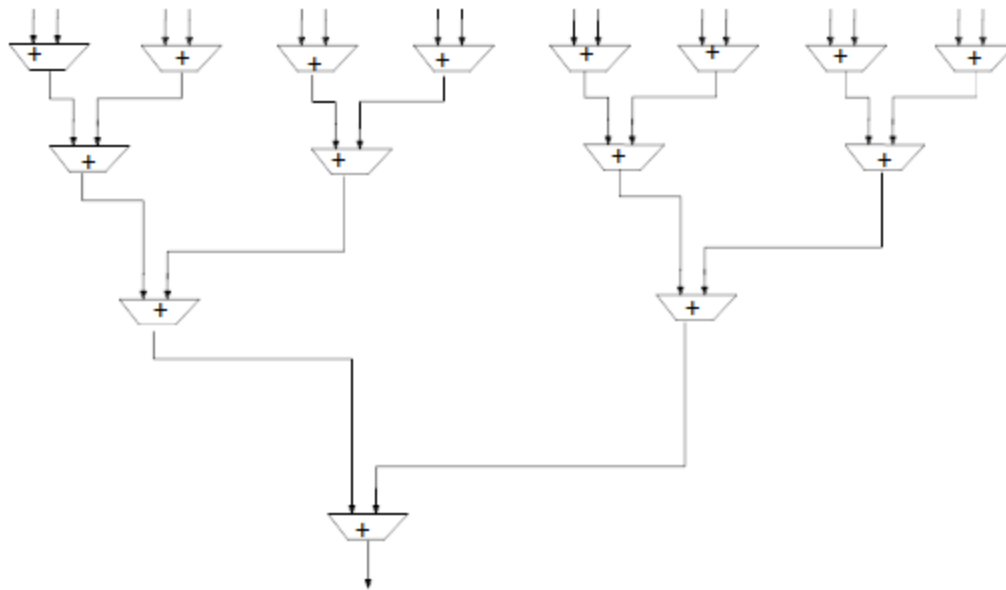
Ανάλυση των πόρων του AD-Census

Είδαμε στο κεφάλαιο 3 ,ότι το AD-Census καταλαμβάνει το μεγαλύτερο μέρος των πόρων. Κάνοντας μια ανάλυση στον AD-Census, διαπιστώσαμε ,όπως φαίνεται και στον πίνακα 4.4, ότι η μονάδα Compute Cost καταλαμβάνει το 94% των LUTs και επομένως εστιάζουμε την προσοχή μας σε αυτήν.

	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
AD Census	22.943 (84 %)	30.219 (55%)	8 (6 %)
Compute Costs Total	21.481 out of 22.943 (94 %)	22.529 out of 30.219 (76 %)	0 out of 8 (0%)
Compute Cost simple Unit	336	353	0
XOR	80 out of 336 (23 %)	0 out of 353 (0%)	0 out of 0 (0%)
Summing	228 out 336 (68 %)	328 out 353 (93 %)	0 out of 0 (0%)
AD-normalized	36 out of 336 (10 %)	9 out of 353 (3%)	0 out of 0 (0%)

Πίνακας 4.4. Αξιοποίηση των πόρων της μονάδας Compute Cost για $D_{max}=64$, $W_c=9$ και $W_a=5$.

Στην εικόνα 4.9 παρουσιάζεται η εσωτερική δομή της μονάδας Compute Cost. Σε αυτή γίνονται 64 παράλληλες λειτουργίες, εκ των οποίων το summing καταλαμβάνει το μεγαλύτερο ποσοστό σε πόρους. Η βασική λειτουργία του είναι το άθροισμα των bits του αποτελέσματος της XOR. Η βασική αυτή λειτουργία έχει υλοποιηθεί με Tree Adders. Η βασική δομή των Tree Adders παρουσιάζεται στην εικόνα 4.10.



Εικόνα 4.10 Βασική δομή των Tree Adders

Στην περίπτωση των Tree Adders, για να προσθέσουμε ένα bitstring των 80 bits, τα προσθέτουμε αρχικά ανά δυο και στη συνέχεια συνεχίζεται η διαδικασία μέχρι να υπολογίσουμε το τελικό άθροισμα.

Κεφάλαιο 5

Υλοποίηση

Γενικά

Ο αλγόριθμος που παρουσιάστηκε δίνει αρκετά καλά αποτελέσματα για διάφορα datasets. Συγκεκριμένα, το μέγιστο ποσοστό των good matches για $W_a=9$, $W_a=5$, $S_{NQ}=64$, $S_{CDQ}=4$ είναι 82,03 %. Στη διάθεση μας όμως έχουμε μια μικρή frga και αυτό μας αποτρέπει από το να χρησιμοποιήσουμε την προηγούμενη υλοποίηση και μας οδηγεί στο να εξετάσουμε νέες αρχιτεκτονικές.

Έχουμε διατηρήσει αυτούσιο το μεγαλύτερο μέρος της αρχιτεκτονικής, όπως αυτή δόθηκε και έχουμε παρέμβει στα σημεία όπου η κατανάλωση των πόρων είναι κρίσιμη. Όπως αναφέραμε στο κεφάλαιο 4, η μονάδα Compute Costs καταλαμβάνει το μεγαλύτερο ποσοστό των πόρων και επομένως εστιάζουμε την προσοχή μας σε αυτήν.

Οι τιμές των παραμέτρων για το Census Window και το aggregation Window ορίζονται σε $W=9$ και $W_a=5$, αντίστοιχα. Το μέγεθος της Neighborhood Queue και της Confident Neighborhood Queue ορίζονται σε $S_{NQ}=64$, $S_{CDQ}=4$.

Reduction Trees

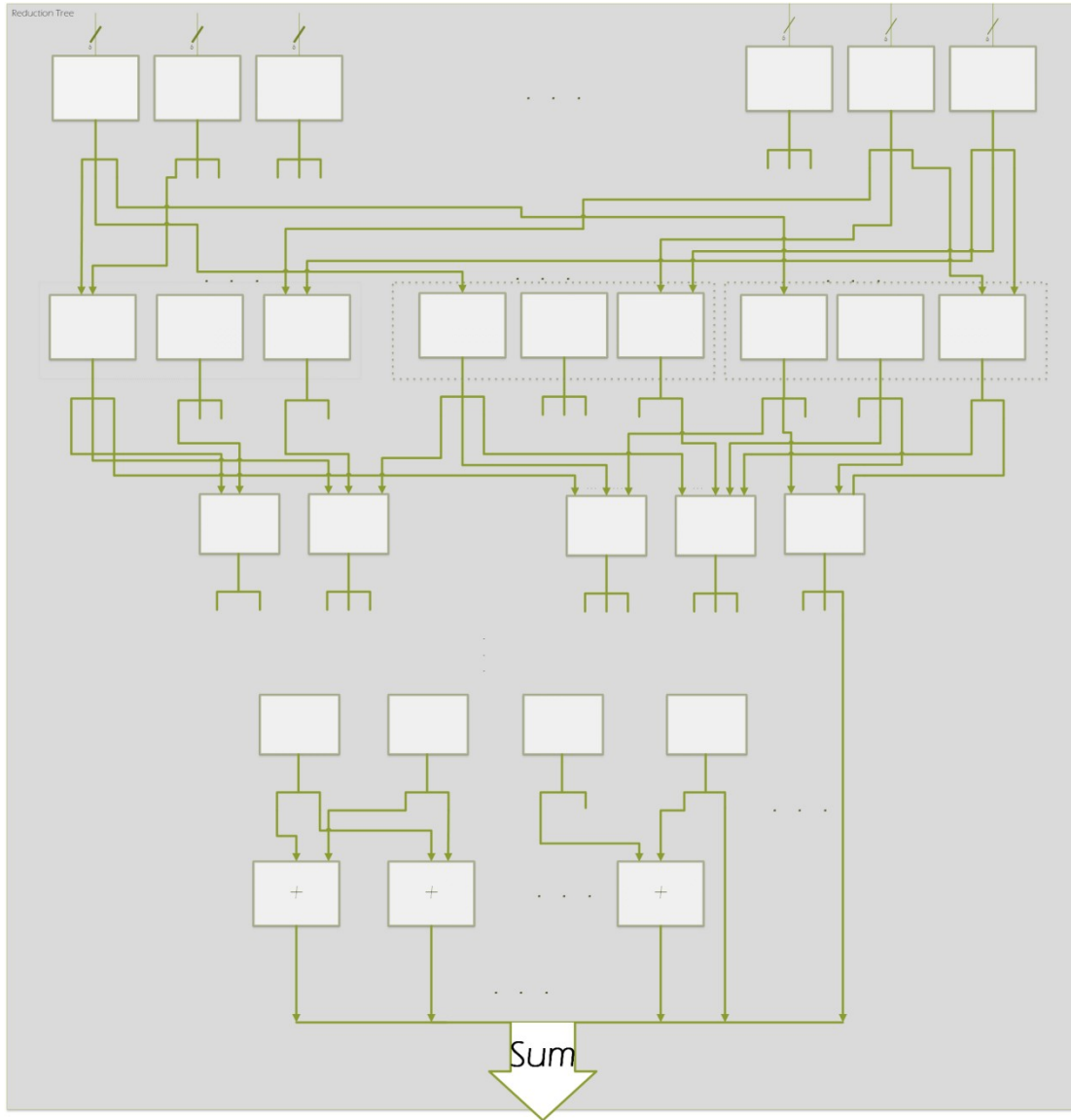
Η μονάδα Summing, στην εικόνα 4.9 του προηγούμενου κεφαλαίου, καταλαμβάνει το μεγαλύτερο ποσοστό των πόρων. Η μονάδα αυτή υλοποιείται με Tree Adders, οι οποίοι αποτελούνται εξολοκλήρου από αθροιστές. Επανασχεδιάσαμε τη συγκεκριμένη μονάδα, χρησιμοποιώντας Reduction Trees.

Η βασική ιδέα είναι ότι ένα bit-string των 80 bits χωρίζεται σε μπλοκς των 6-bit, τα οποία τροφοδοτούν τις μονάδες του Reduction Tree. Η κάθε μονάδα είναι ένας κωδικοποιητής. Η κάθε μονάδα είναι ένας κωδικοποιητής η οποία δέχεται μέχρι 6 εισόδους και μετασχηματίζει την είσοδο στον αριθμός των '1' που περιέχει. Με απλά λόγια, προσδιορίζει κάθε φορά πόσοι είναι οι '1'. Για παράδειγμα, αν υποθέσουμε ότι η βασική μονάδα δέχεται ως είσοδο το bit-string {1,1,0,1,0,1} τότε θα πάρουμε στην έξοδο '100', η οποία αποτελεί την δυαδική τιμή των '1', στη συγκεκριμένη περίπτωση 4.

Τα bits του προηγούμενου βήματος, ομαδοποιούνται με βάση τη θέση τους σε κάθε μονάδα (LSB, MSB κτλ). Τα bits εισέρχονται με τη σειρά τους σε νέες μονάδες στο επόμενο επίπεδο. Οι έξοδοι από τις μονάδες, στη συνέχεια ομαδοποιούνται σε σήματα συγκεκριμένης τάξης μεγέθους, τα οποία εισέρχονται και αυτά με τη σειρά τους σε νέες μονάδες. Σε κάθε επίπεδο, από ένα σημείο και μετά, είναι έτοιμο ένα bit του συνολικού

αθροίσματος. Το δέντρο δεν συνεχίζει, μέχρι να υπολογίσει όλα τα bits ένα-ένα, αλλά στο πέμπτο επίπεδο αντί για τις βασικές μονάδες, χρησιμοποιούνται 1-bit αθροιστές, οι οποίοι μας δίνουν απευθείας τα υπολειπόμενα bits. Σε κάθε επίπεδο οι βασικές μονάδες μειώνονται.

Το γενικό διάγραμμα παρουσιάζεται στην εικόνα 5.1. Ανάμεσα σε κάθε επίπεδο έχουν τοποθετηθεί pipeline registers. Στην εικόνα 5.2, δείχνουμε ένα παράδειγμα ενός 48-bit bitstring. Το bitstring χωρίζεται σε 8 ομάδες των 6 bits, τα οποία διέρχονται αρχικά από τις βασικές μονάδες. Στο πρώτο επίπεδο έχουμε τρεις τάξεις μεγέθους και χρειαζόμαστε 6 βασικές μονάδες. Τα σήματα με το ίδιο χρώμα ανήκουν στην ίδια τάξη μεγέθους. Στο δεύτερο και στο τρίτο επίπεδο χρειαζόμαστε τέσσερις βασικές μονάδες, ενώ στο τέταρτο τρεις. Το αποτέλεσμα που προκύπτει στο πέμπτο επίπεδο, οδηγείται σε τρεις αθροιστές του 1-bit, όπου το κρατούμενο διαδίδεται από τον έναν στον άλλο ή αλλιώς σε έναν 3-bit αθροιστή.



Εικόνα 5.1 . Reduction Tree

Εφαρμογές του Reduction Tree.

Το Reduction Tree, γενικότερα, μπορεί να εφαρμοστεί στις περιπτώσεις όπου απαιτούνται η προσθέσεις των x bits . Βέβαια, στις περιπτώσεις αυτές ($x > 1$) δεν χρειάζεται η αρχική ομαδοποίηση και τα bitstrings των αριθμών μπορούν να οδηγηθούν απευθείας στο επίπεδο 1.

Εφαρμόζουμε, επομένως, την ίδια μεθοδολογία και στο κομμάτι του aggregation unit, στην οποία γίνονται D_{max} παράλληλες προσθέσεις, που αποτελούν τα Vertical Sums. Συγκεκριμένα, γίνονται 5 προσθέσεις των 6 bits. Σε αυτή την περίπτωση, απαιτούνται 6 βασικές μονάδες στο πρώτο επίπεδο και το δέντρο έχει τρία επίπεδα και όχι 5.

Στερεοσκοπικές κάμερες

Χρησιμοποιούμε το VmodCam Board, το οποίο έχει δυο Aptina MT9D112 2-Megapixel CMOS image sensors. Το σύστημα παρέχει 30 fps και εικόνες με ανάλυση 640x480. Ωστόσο, η μέγιστη ανάλυση που μπορούμε να πάρουμε από τις κάμερες είναι 1600x1200. Επιπλέον, παρέχει ένα image flow processor και επιτρέπει διάφορες μορφές (formats) των εικόνων. Οι κάμερες ,πριν χρησιμοποιηθούν, θα πρέπει να διαμορφωθούν κατάλληλα.

Επιλογή του format

Λόγω της φύσης του αλγορίθμου, χρειαζόμαστε grayscale εικόνες. Με κατάλληλη διαμόρφωση των καμερών, είχαμε τη δυνατότητα να πάρουμε μονοχρωματικές εικόνες. Το μέγεθος του pixel, σε αυτή την περίπτωση, είναι 16 bit , από τα οποία επιλέγουμε τα 8. Στην εικόνα 5.2, δείχνουμε τις δυο εικόνες όπως αυτές προκύπτουν από τις κάμερες. Στην εικόνες αυτές, παρατηρούμε ένα flickering το οποίο αλλοιώνει την ποιότητά τους, γεγονός που ήταν ακόμη πιο έντονο στο disparity map.

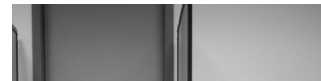


Left Image

Right Image

Εικόνα 5.3 Μονοχρωματικές εικόνες

Το σύστημα των καμερών, ωστόσο, μας παρέχει επιπλέον τη δυνατότητα να πάρουμε εικόνες σε YUV format, από την οποία μορφή επιλέγουμε την Y συνιστώσα . Στην εικόνα 5.4, δείχνουμε τις δύο εικόνες. Το flickering δεν εμφανίζεται πλέον και η διαφορά στην ποιότητά τους είναι αισθητή.



Left

Right

Εικόνα 5.4 Βελτιωμένες εικόνες.

Διόρθωση εικόνας.

Ο αλγόριθμος θεωρεί ότι η δυο εικόνες είναι rectified ,ωστόσο δεν πραγματοποιήθηκε rectification . Εξετάσαμε, όμως, τις δύο εικόνες και παρατηρήσαμε ότι έχουν μια διαφορά στον κάθετο άξονα. Στην εικόνα 5.6, προβάλλουμε τις εικόνες που παρουσιάζονται στην εικόνα 5.5, τη μια τοποθετημένη πάνω στην άλλη. Η διαφορά στον κάθετο άξονα είναι εμφανής. Η διαφορά αυτή μετρήθηκε και είναι 19 γραμμές. Διορθώνουμε τις δύο εικόνες, ώστε να εξαλείψουμε τη διαφορά και στη συνέχεια προβάλλουμε ξανά τις δυο εικόνες (Εικόνα 5.7). Η μόνη διαφορά που προκύπτει είναι στον οριζόντιο άξονα, η οποία είναι και επιθυμητή. Στη συνέχεια, μετακινήσαμε την αριστερή εικόνα κατά 64 pixel προς τα δεξιά και όπως φαίνεται στην εικόνα 5.7, οι δύο εικόνες συμπίπτουν στα σημεία όπου προβάλλουν τον ίδιο χώρο.

Η διαφορά των 19 γραμμών βρέθηκε εξετάζοντας το σφάλμα που προκύπτει από την αφαίρεση των δύο εικόνων, θεωρώντας κάθε φορά διαφορετικό αριθμό γραμμών και επιλέγοντας εκείνο με το μικρότερο σφάλμα. Στην επιλογή αυτή, εξετάστηκαν διαφορετικές εικόνες, στις οποίες το σύστημα των καμερών ήταν τοποθετημένο σε διαφορετική θέση .

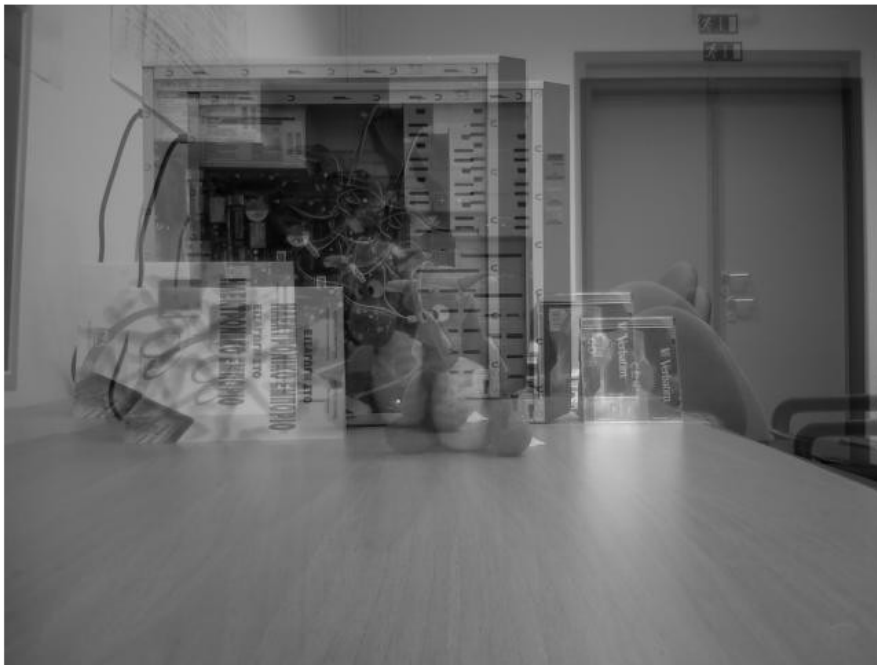


Left



Right

Εικόνα 5.5 Αρχικές εικόνες



Εικόνα 5.6



Εικόνα 5.7

Για να εξασφαλίσουμε επομένως στο σύστημα μας ότι οι δύο εικόνες θα διαφέρουν μόνο στον οριζόντιο άξονα, παραλείπουμε τις 19 πρώτες γραμμές της αριστερής εικόνας και τις 19 τελευταίες γραμμές της δεξιάς εικόνας.

Διαμόρφωση κάμερών.

Διαμορφώνουμε τις κάμερες ώστε να μας παρέχουν video ανάλυσης 640x480 και YUV format. Ενεργοποιήσαμε, επιπλέον, ενέργειες, όπως Auto Exposure και Auto White Balance . Όλες οι απαιτούμενες ενέργειες για τη διαμόρφωση των καμερών γίνονται στη μονάδα Camera Controller ,όπου η κάθε κάμερα διαμορφώνεται ξεχωριστά.

Σύστημα πραγματικού χρόνου.

Το διάγραμμα ολόκληρου του συστήματος παρουσιάζεται στην εικόνα 5.8.

To disparity map αποθηκεύεται στη μνήμη, όπου και στέλνεται στη συνέχεια μέσω HDMI στην οθόνη.

Μετά από μια αρχική καθυστέρηση, το σύστημα μπαίνει σε λειτουργία και παράγει frames σε συνεχή βάση. Για την υλοποίηση των μονάδων Camera Controller, HDMI Controller, χρησιμοποιήσαμε τα διάφορα παραδείγματα που υπάρχουν στο [21]. Βέβαια, όλα έχουν παραμετροποιηθεί ως προς την ανάλυση.

Οργάνωση της μνήμης

Έχουμε χωρίσει την μνήμη σε τρεις περιοχές, μια για την κάθε εικόνα. Το μέγεθος της κάθε περιοχής είναι παραμετροποιημένο, αρκεί βέβαια να επαρκεί για το μέγεθος της εικόνας. Μία εικόνα, στην οποία τα pixel είναι 8 bits, χρειάζεται $M \times N$ bytes. Επομένως η περιοχή θα πρέπει να είναι τουλάχιστον $M \times N$ bytes. Μέσω του Xilinx Memory Interface Generator, ορίσαμε 6 θύρες για την DDR2. Οι τρεις χρησιμοποιούνται για εγγραφή και οι άλλες τρεις για ανάγνωση, συγκεκριμένα δυο θύρες χρησιμοποιούνται για εγγραφή και ανάγνωση σε μια περιοχή της μνήμης.

Write Controller

Κάθε φορά, γράφουμε στη μνήμη W λέξεις των 32 bit. Το W εξαρτάται από το ποιά εικόνα θα γραφτεί. Για τις εικόνες που έρχονται από τις κάμερες, η συγκεκριμένη παράμετρος είναι 22, ενώ για την εικόνα που έρχεται από το Stereo Vision Module είναι 20. Η συγκεκριμένη μονάδα εξασφαλίζει ότι έχουν γραφτεί οι συγκεκριμένες λέξεις στην εσωτερική FIFO της DDR2, πριν ζητήσει να γραφτούν όλα στη μνήμη. Επιπλέον, ορίζει την διεύθυνση στην οποία θα γραφτούν τα δεδομένα και εξετάζει αν η εικόνα έχει γραφτεί στη μνήμη, ώστε να ορίσει ξανά την αρχική διεύθυνση, στην οποία θα γραφτεί η επόμενη εικόνα.

Arrange Write Data

Το μέγεθος του pixel είναι 8 bits, όμως στην εσωτερική FIFO της DDR γράφουμε λέξεις των 32 bits, επομένως η συγκεκριμένη μονάδα καθυστερεί τα pixel που έρχονται, ώστε να σταλούν ανά τετράδες στην FIFO. Ουσιαστικά γράφει κάθε φορά στη FIFO 4 pixels.

Read Controller

Η μονάδα αυτή ζητάει από τη DDR να μας στείλει δεδομένα και ορίζει και τις διευθύνσεις των δεδομένων που χρειαζόμαστε. Εξετάζει αν διαβάσαμε 64 λέξεις από την εσωτερική FIFO της συγκεκριμένης θύρας,. Ελέγχει,επιπλέον,αν διαβάσαμε ολόκληρη την εικόνα, ώστε να αρχικοποιήσουμε ξανά τις διευθύνσεις.

Arrange Read Data

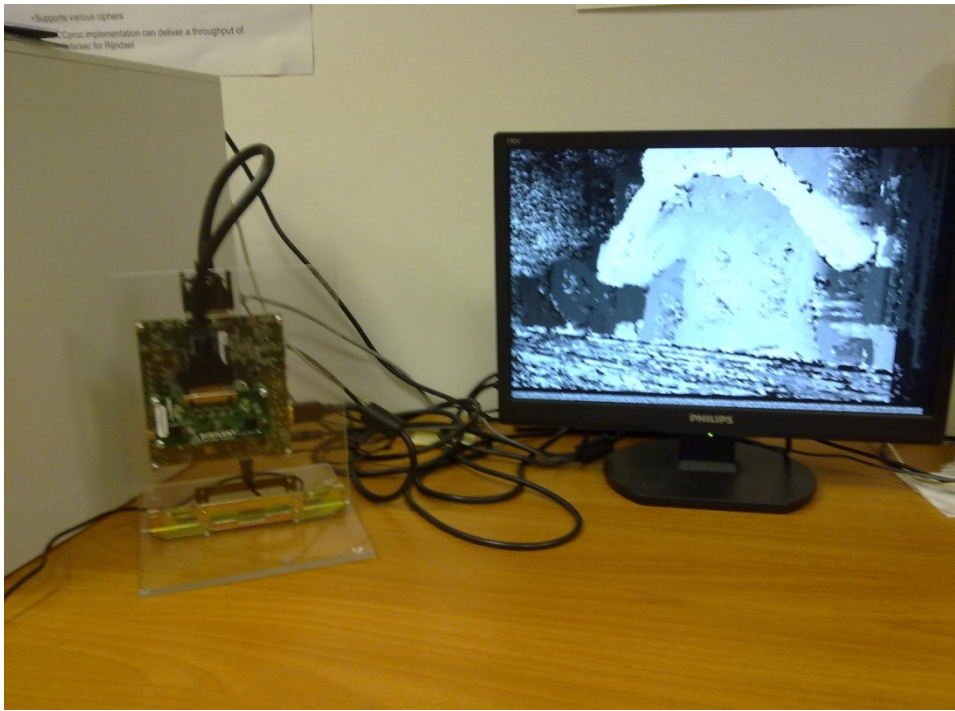
Τα δεδομένα που έρχονται από την DDR γράφονται σε μια μικρή FIFO, η οποία χρησιμεύει ως Buffer. Η μονάδα Arrange Read Data εξασφαλίζει ότι θα στέλνεται ένα pixel τη φορά, όταν τις ζητείται.

Αποτελέσματα

Το υλικό σύστημα, όπως χρησιμοποιήθηκε, παρουσιάζεται στην επόμενη εικόνα, ενώ στην εικόνα 5.10 παρουσιάζουμε ένα στιγμιότυπο του συστήματος. Το disparity Map παρουσιάζεται στην εικόνα 5.11 μαζί με τις αρχικές εικόνες

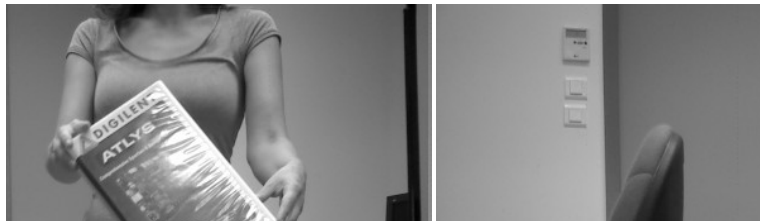


Εικόνα 5.9 Υλικό σύστημα



Εικόνα 5.10 .Στιγμιότυπο του συστήματος

Εικόνα 5.11



Αρχικές εικόνες



Εικόνα 5.12 Disparity Map

Ανάλυση των πόρων

Η αξιοποίηση των πόρων, μετά τις αλλαγές, παρουσιάζεται στον πίνακα 5.1, για την μονάδα Compute Costs. Όπως βλέπουμε, είχαμε μια σημαντική μείωση της τάξης των 7.417 σε LUTs και 10.521 σε Registers. Στον πίνακα 5.2, παρουσιάζουμε τα συγκεντρωτικά αποτελέσματα πριν και μετά τις βελτιώσεις, για το κομμάτι του AD-Census. Όπως προκύπτει, μειώσαμε τα LUTs κατά 33 % και τους Registers κατά 35%. Βεβαία, το κυρίως πρόβλημα παρουσιάζεται στην έλλειψη των LUTs. Στον πίνακα 5.2, παρουσιάζουμε την αξιοποίηση των πόρων για το συνολικό κομμάτι του StereoVision Module, για ανάλυση 640x480 , Dmax=64 , W =9 και Wa =5 . Όπως φαίνεται στον πίνακα, η βελτίωση αυτή εξακολουθεί να μην επαρκεί, καθώς απαιτεί το 101% των LUTs, χωρίς να λάβουμε υπόψη και τους πόρους που χρειάζονται για το συνολικό σύστημα.

Στην εικόνα 5.8 φαίνονται οι διαφορές σε LUTs του AD-Census για τις διάφορες τιμές του disparity, πριν και μετά τις αλλαγές.

	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
AD Census	15.526	19.698	8
Compute Costs Total	14.145 out of 15.526 (91 %)	12.417 out of 19.698 (63 %)	0 out of 8 (0%)
Compute Cost simple Unit	192	48	0
XOR	80 out of 192 (42 %)	0 out of 48 (0%)	0 out of 0 (0%)
Summing	76 out 192 (39 %)	39 out 48 (81 %)	0 out of 0 (0%)
AD-normalized	36 out of 192 (19 %)	9 out of 48 (19%)	0 out of 0 (0%)

Πίνακας 5.1 Αξιοποίηση των πόρων για τη μονάδα Compute Costs, μετά τις αλλαγές.

W=9	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
AD Census Optimized	15.526	19.698	8
AD Census UnOptimized	22.943	30.219	8
Optimization(%)	33%	35%	0%

Πίνακας 5.2 . Σύγκριση των αποτελεσμάτων.

	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total Consumed	27.674 (101 %)	29.840 (54%)	74 (63 %)
AD Census	15.526	19.698	8
Aggregation	8.303	6.449	66
Left/Right Check	4.345	3.459	0
Scanline Belief Propagation	443	449	0

Πίνακας 5.3 Αξιοποίηση των πόρων μετά τις αλλαγές για $D_{max}=64$.

Εικόνα 5.8 . Σύγκριση των LUTs του AD-Census για διάφορες τιμές του disparity

Εφαρμόζοντας τις αλλαγές και στο aggregation unit, η αξιοποίηση των πόρων παρουσιάζεται στον πίνακα 5.9. Οι διαφορές, σε αυτή την περίπτωση, είναι πολύ μικρές και ο λόγος είναι ότι ο αριθμός των bitstring που προστίθενται είναι μικρός.

Dmax =64	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total Consumed	27.275(99 %)	29.569 (54%)	74 (63 %)
AD Census	15.526	19.698	8
Aggregation	7.834	5.236	66
Left/Right Check	4.345	3.459	0
Scanline Belief Propagation	443	449	0

Πίνακας 5.3 Αξιοποίηση των πόρων μετά τις αλλαγές και στο aggregation unit.

Αν και η καλύτερη περίπτωση για μια 640x480 ανάλυση θα ήταν να έχουμε disparity 64, κάτι τέτοιο δεν είναι εφικτό . Επομένως, θυσιάζουμε ένα μέρος της ελάχιστης απόστασης που μπορούν να εντοπίσουν οι κάμερες και μειώνουμε το disparity, για να έχουμε το σύστημα πραγματικού χρόνου. Το σύστημα λειτουργεί πλήρως για Dmax =54 για τη συγκεκριμένη ανάλυση ,ενώ για ανάλυση 400x320 μπορούμε να έχουμε Dmax=57.

Στους πίνακες 5.4, 5.5 και 5.6 παρουσιάζονται οι πόροι που απαιτούνται για διάφορες τιμές του disparity, για τις οποίες έχουμε πετύχει place and route.

Dmax =40	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total Consumed	17.251 (63 %)	19.255 (35 %)	70 (60 %)
AD Census	9.935 /17.251 (58%)	12.784/19.255 (66%)	28/70 (40%)
Aggregation	4.695 /17.251 (27%)	3.605/19.255 (19%)	42/70 (60%)
Left/Right Check	2.486 /17.251 (14%)	2.454/19.255 (13%)	0/70 (%)
Scanline Belief Propagation	274/17.251 (2%)	281/19.255 (2%)	0/70 (%)

Dmax=50	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total Consumed	21.537 (78%)	23.428 (42 %)	87
AD Census	12.472/21.537 (58 %)	15.584/23.428 (67 %)	33/87 (38%)
Aggregation	5.929/21.537 (28%)	4.505/23.428 (19 %)	54/87 (62%)
Left/Right Check	3.311/21.537 (15%)	2.871/23.428 (12%)	0/87 (0 %)
Scanline Belief Propagation	309/21.537 (1%)	316/23.428 (1 %)	0/87 (0 %)
Dmax=54	LUTs (%)	Flip-Flops (%)	BRAMs (%)
Available	27.288	54.576	116
Total Consumed	22.942 (84 %)	25.102 (45 %)	89
AD Census	13.041/22.942 (57 %)	16.704 /25.102 (67 %)	35/89 (39%)
Aggregation	6.255 /22.942 (27%)	4.845 /25.102 (19%)	54/89 (61%)
Left/Right Check	3.617/22.942 (16%)	3.039/25.102 (12%)	0/89 (0%)
Scanline Belief Propagation	344/22.942 (2 %)	351 /25.102 (1 %)	0/89 (0%)

Πίνακας 5.6 Αξιοποίηση των πόρων

Στον πίνακα 5.7, παρουσιάζουμε τους πόρους που απαιτούνται για το συνολικό σύστημα για διάφορες τιμές του disparity. Όπως ήδη έχουμε αναλύσει, η αύξηση του Dmax οδηγεί στην αύξηση των πόρων.

Dmax	LUTs (%)	Flip-Flops (%)	BRAMs (%)
20	11.026 (40 %)	11.363 (20%)	58 (50 %)
30	15.241 (55 %)	15.777 (28 %)	59 (51 %)
35	17.339 (63 %)	18.212 (33 %)	68 (59 %)
40	19.421 (71 %)	20.574 (37 %)	76 (67 %)
45	21.525 (78 %)	22.684 (41 %)	84(73 %)
50	23.529 (86 %)	24.847 (45 %)	68 (59 %)
54	25.752 (94 %)	27.123 (49 %)	68 (59 %)

Πίνακας 5.7 . Αξιοποίηση των πόρων του συνολικού συστήματος για διάφορες τιμές του disparity

Ανάλυση της απόδοσης

Το σύστημά μας παράγει 30 fps, λόγω των περιορισμών που εισάγουν οι κάμερες. Ωστόσο, ο αλγόριθμος έχει τη δυνατότητα να παράγει ένα αποτέλεσμα ανά κύκλο. Στον πίνακα 5.8, παρουσιάζουμε το θεωρητικό μέγιστο ρολόι της προηγούμενης υλοποίησης και τις δικής μας. Παρατηρούμε μεγάλη πτώση στη συχνότητα λειτουργίας, η οποία οφείλεται και στη χρήση διαφορετικής frga, αυτή που χρησιμοποιούμε είναι μικρότερη σε σχέση με την προηγούμενη ,αλλά και στο γεγονός ότι ορίσαμε τις επιλογές των εργαλείων ,ώστε να βελτιώσουν τη σχεδίαση ως προς το χώρο και όχι ως προς το χρόνο.

	Μέγιστη Συχνότητα
Προηγούμενη υλοποίηση	201.518 MHz
Δική μας υλοποίηση	69.788 MHz

Πίνακας 5.8 Συχνότητα λειτουργίας στις δυο υλοποιήσεις.

Η θεωρητική απόδοση του συστήματος μετρούμενη σε fps, για διαφορετικές αναλύσεις, παρουσιάζονται στον παρακάτω πίνακα.

Ανάλυση	100x83	640x480	1024x853	1600x1200
Απόδοση(fps)	8.408	227	79.8	36

Πίνακας 5.9 Μέγιστο θεωρητικό Throughput.

Κεφάλαιο 6

Επιβεβαίωση Λειτουργίας

Έχοντας πραγματικές εικόνες στη διάθεση μας, προσπαθήσαμε να επιβεβαιώσουμε τη λειτουργία του συστήματός μας, συγκρίνοντας την πραγματική απόσταση των αντικειμένων από την κάμερα με αυτή που προκύπτει με βάση το disparity map. Για τον

$$Depth = \frac{b * f}{d * DP_{hor}}$$

υπολογισμό των αποστάσεων χρησιμοποιούμε την σχέση $Depth = \frac{b * f}{d * DP_{hor}}$, όπου το d είναι οι τιμές που προκύπτουν από το disparity map.

Η διαδικασία αυτή πραγματοποιήθηκε σε software, στην οποία υπολογίζουμε για κάθε σημείο της εικόνας την απόσταση. Μετρήσαμε, στη συνέχεια, με ένα απλό μέτρο την απόσταση των αντικειμένων από την κάμερα. Η διαδικασία αυτή πραγματοποιήθηκε για πολλές εικόνες και η απόσταση που υπολογίζεται διαφέρει από την πραγματική απόσταση από 8 μέχρι 10 εκατοστά. Οι διαφορές αυτές οφείλονται στο γεγονός ότι δεν έχει πραγματοποιηθεί rectification στις δυο εικόνες, επομένως οι τιμές που προέκυψαν για το disparity έχουν μια μικρή απόκλιση.

Ο αλγόριθμος, επιπλέον, έχει παρουσιασθεί σε simulation με την ίδια διαδικασία που πραγματοποιήθηκε και στην προηγούμενη διπλωματική για μικρά data_sets και συγκρίθηκαν με τα αποτελέσματα του αλγορίθμου πριν τις αλλαγές. Δεν εντοπίστηκαν διαφορές, κάτι το οποίο είναι αναμενόμενο, εφόσον δεν έχει αλλάξει η λειτουργικότητα του αλγορίθμου.

Για την παραπάνω διαδικασία χρησιμοποιήθηκε το σύστημα πραγματικού χρόνου, όπως φαίνεται στην εικόνα 5.3, με τη διαφορά όμως ότι το disparity map στέλνεται σε έναν υπολογιστή. Για την λειτουργία αυτή υλοποιήθηκε στην fpga το Communication Module, το οποίο διαβάζει την εικόνα από την DDR και τη στέλνει στη USB θύρα, όταν αυτή το ζητήσει. Στη θύρα USB στέλνεται κάθε φορά ένα byte, επομένως 1 pixel. Υλοποιήθηκε, επιπλέον, σε software ένα πρόγραμμα, το οποίο ζητάει από την fpga την εικόνα, την οποία στη συνέχεια αποθηκεύει σε png format. Για την υλοποίηση του προγράμματος χρησιμοποιήσαμε τη βιβλιοθήκη libfpgalink [21], η οποία προσφέρει δυνατότητες εγγραφής και ανάγνωσης στο USB, καθώς και δυνατότητες προγραμματισμού της FPGA.

Στην εικόνα 6.1, παρουσιάζουμε την επικοινωνία μεταξύ του υπολογιστή και της FPGA.



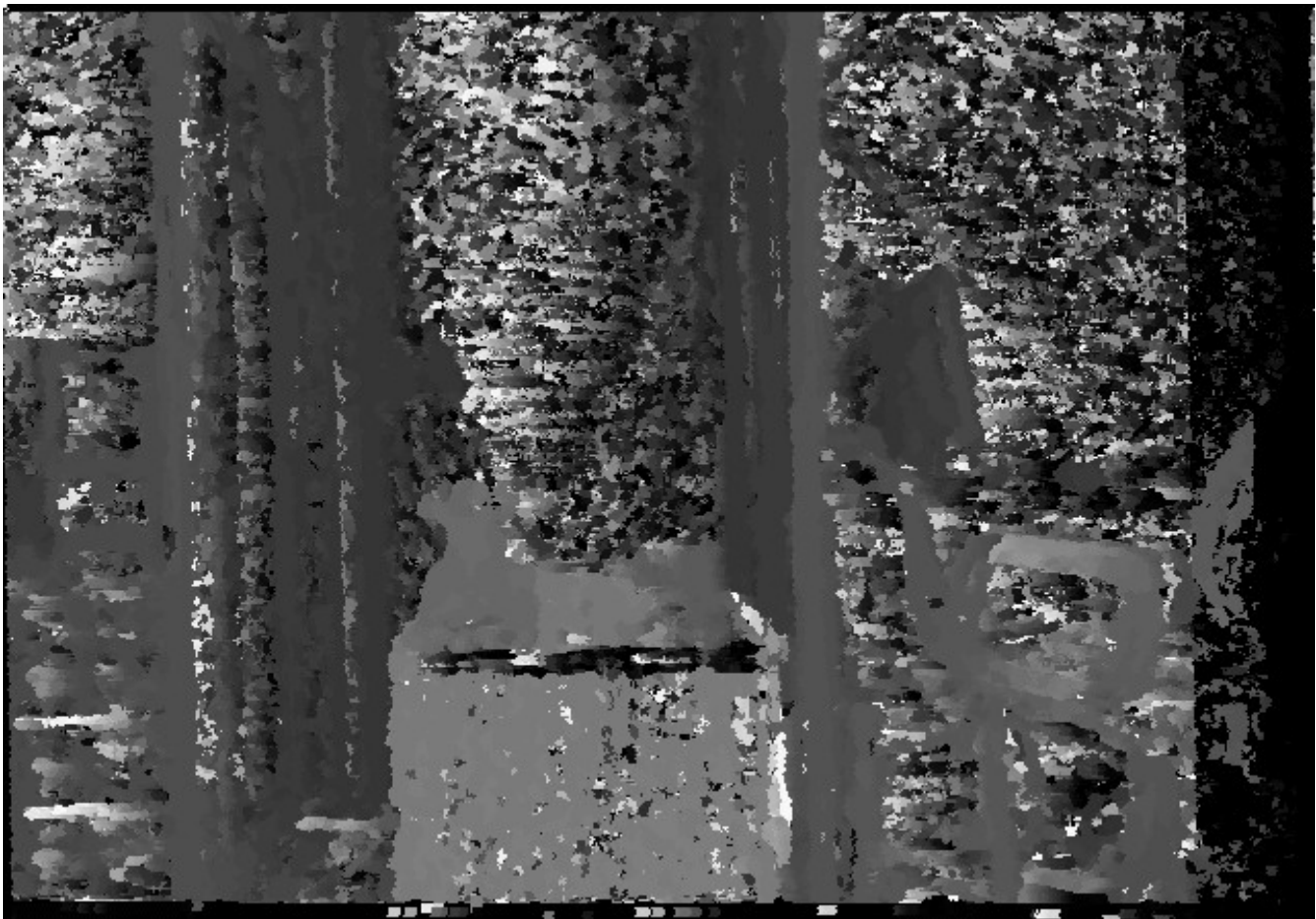
Εικόνα 6.1 Επικοινωνία FPGA –Υπολογιστή.

Τα αποτελέσματα της πρώτης μεθοδολογίας, παρουσιάζονται παρακάτω . Το εύρος τιμών των pixel του disparity map που παρουσιάζεται είναι [0 216], ωστόσο για τον υπολογισμό των αποστάσεων χρησιμοποιήθηκαν οι πραγματικές τιμές, οι οποίες έχουν εύρος [0 53].



Left

Right



Disparity map

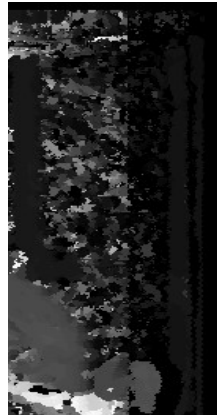
Απόσταση διαφόρων αντικειμένων στην εικόνα.



Left



Right



Disparity map



Απόσταση διαφόρων αντικειμένων στην εικόνα.

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές επεκτάσεις

Σε αυτή την διπλωματική εργασία είδαμε πως μια προηγούμενη αρχιτεκτονική βελτιώθηκε ώστε να καταναλώνει 33% λιγότερους πόρους. Για να μπορέσει να χωρέσει η σχεδίαση σε μια μικρή FPGA έπρεπε να μειωθεί και το εύρος του disparity.

Το σύστημα αυτό, θα μπορούσε να βελτιωθεί εφαρμόζοντας rectification στις εικόνες που έρχονται από τις κάμερες, ώστε να βελτιωθεί η ποιότητά του disparity map. Επιπλέον, θα πρέπει να εξεταστούν δυνατότητες χρήσης της DDR2 , για την μείωση του υπολογιστικού κόστους.

Βιβλιογραφία

- [1] N. K. Ratha, A. K. Jain, "Computer Vision Algorithms on Reconfigurable Logic Arrays", *IEEE Transactions on Parallel and Distributed Systems*, Jan. 1999, Vol. 10, No. 1, pp. 29-43.
- [2] C. Murphy, D. Lindquist, A. M. Rynning, T. Cecil, S. Leavitt and M. L. Chang , "Low-Cost Stereo Vision on an FPGA", in *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2007, pp. 333-334.
- [3] K. Ratnayake, A. Amer, "An FPGA-Based Implementation of Spatio-Temporal Object Segmentation", in *IEEE International Conference on Image Processing (ICIP)*, 2006, pp.3265-3268.
- [4] K. Konolige, "Small Vision Systems: Hardware and Implementation", in *International Symposium on Robotics Research*, 1997, pp. 111-116.
- [5] B. Rajan, S.Ravi, "FPGA Based Hardware Implementation of Image Filter With Dynamic Reconfiguration Architecture", in *IJCSNS International Journal of Computer Science and Network Security*, Dec. 2006, Vol. 6, No. 12, pp. 121-127.
- [6] D. K. Masrani, W. J. MacLean, "A Real-Time Large Disparity Range Stereo-System using FPGAs", in *Proceedings of the IEEE International Conference on Computer Vision Systems*, 2006, pp. 42-51.
- [7] C. Claus, A. Laikat, L. Jia, W. Stechele, "High performance FPGA based optical flow calculation using the census transformation", in *IEEE Intelligent Vehicles Symposium*, 2009, pp.1185-1190.
- [8] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim, J. W. Jeon, "FPGA Design and Implementation of a Real-Time Stereo Vision System", in *IEEE Transactions on Circuits and Systems for Video Technology*, Jan. 2010, Vol. 20, No. 1, pp. 15-26.
- [9] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems", article in *Computer Vision and Image Understanding*, Mar. 2010. [Online]. Available: www.elsevier.com/locate/cviu

- [10] S. Hadjitheophanous, C. Ttofis, A. S. Georghiades, T. Theocharides, "Towards Hardware Stereoscopic 3D Reconstruction, A Real-Time FPGA Computation of the Disparity Map", in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar 2010, pp. 1743-1748.
- [11] D. Scharstein, R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", in *International Journal of Computer Vision*, Apr. 2002, Vol.47, No. 1-3, pp. 7-42.
- [12] J. Salmen, M. Schlipf, J. Edelbrunner, S. Hegemann, S. Lücke, "Stereo Vision: Making more out of Dynamic Programming", in *Computer Analysis of Images and Patterns*, 2009, Vol. 5702/2009, pp. 1096-1103.
- [13] R. Zabih, J. Woodfill, "Non-parametric local transforms for computing visual correspondence", in *Proceedings of the third European conference on Computer Vision (ECCV)*, 1994, Secaucus, NJ, USA: Springer-Verlag New York, Inc., pp. 151-158.
- [14] G. Xiong, X. Li, H. Chen, D. Lee, "Color Rank and Census Transforms using Perceptual Color Contrast", in *International Conference on Control, Automation, Robotics and Vision(ICARCV)*, Dec. 2010, pp. 1225-1230.
- [15] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, M. Gelautz, "Fast Cost-Volume Filtering for Visual Correspondence and Beyond", in *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [16] Stefano Mattoccia, "Stereo vision: algorithms and applications", VIALAB Bologna, November 2011. <http://www.vision.deis.unibo.it/smatt/stereo.htm>
- [17] JIN, Minxi; MARUYAMA, Tsutomu. A real-time stereo vision system using a tree-structured dynamic programming on FPGA. In: *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*. ACM, 2012. p. 21-24.
- [18] JIN, Minxi; MARUYAMA, Tsutomu. A fast and high quality stereo matching algorithm on FPGA. In: *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*. IEEE, 2012. p. 507-510.
- [19] TTOFIS, Christos, et al. Edge-directed hardware architecture for real-time disparity map computation. 2012.
- [20] JIN, Minxi; MARUYAMA, Tsutomu. A fast and high quality stereo matching algorithm on FPGA. In: *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*. IEEE, 2012. p. 507-510.
- [21] <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,719,932&Prod=VMOD-CAM>

[22] <http://www.makestuff.eu/wordpress/software/fpgalink/>