Στερεοσκοπική όραση με χρήση νευρωνικού δικτύου

Βασίλης Γκολέμης
Επιβλέπων: Αναστάσιος Ντελόπουλος

Ομάδα κατανόησης πολυμέσων
Εργαστήριο Επεξεργασίας Πληροφορίας
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

3 Νοεμβρίου 2017

Κατανόηση $3D$ χώρου

Γιατί;

- Πλοήγηση (ρομποτική, αυτόνομη οδήγηση, ιατρική)
- Αλληλεπίδραση (αυτοματοποίηση παραγωγής)
- Καταγραφή (δημιουργία χάρτη από αεροφωτογραφία)

Πως;

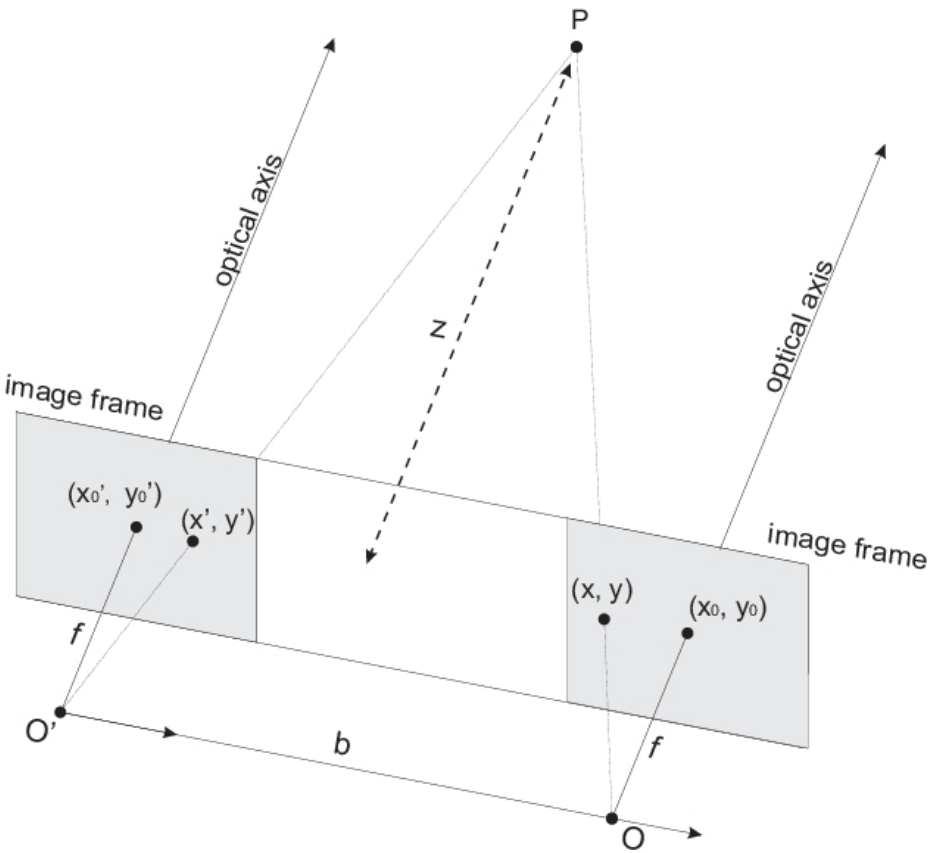
- Lidar
- Όραση (monocular, stereo, multiview)
- Δομημένο φως (kinect)

Στερεοσκοπική Γεωμετρία

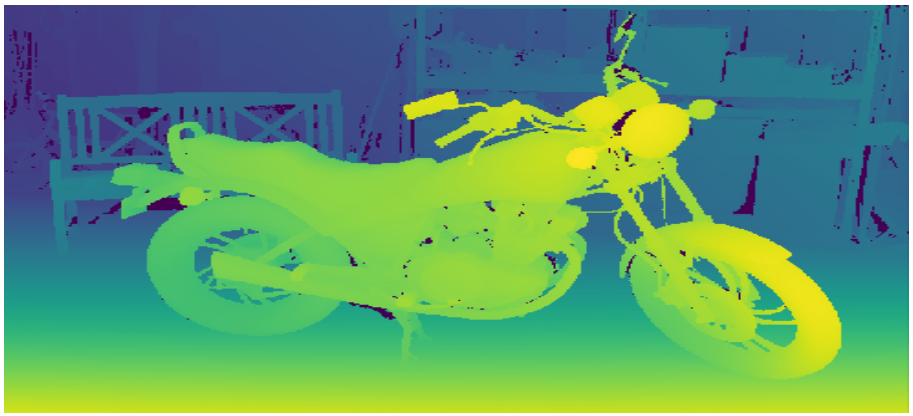
- $P = (X, Y, Z)$
- $y^{'} = y$ (στερεοσκοπικός περιορισμός)
- $x^{'} = f\frac{X}{Z}$
- $x = f\frac{X - b}{Z}$
- $Z = \frac{fb}{d}$
- $d = x^{'} - x$ (παράλλαξη)

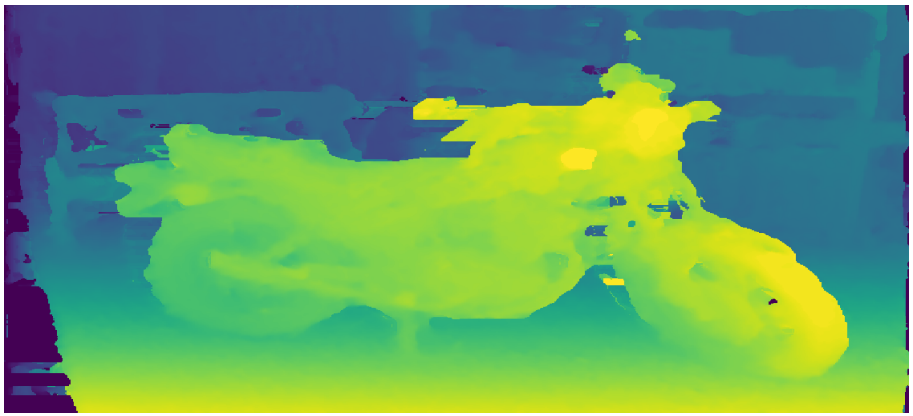
Στόχος:

- $\forall p \in I_L$ βρες d



Χάρτης παράλλαξης



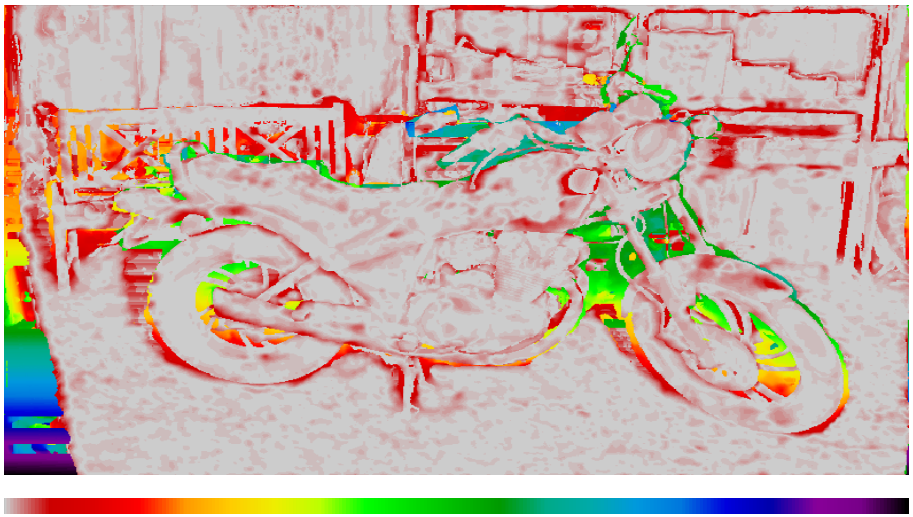


(α′) $\mathbf{D}^{L}_{\mathbf{ground\_truth}}$

(β′) $\mathbf{D}^{L}_{\mathbf{predicted}}$

Αξιολόγηση χάρτη παράλλαξης

- Απόλυτο σφάλμα πρόβλεψης $\rightarrow$ μέση τιμή ($2.058px$)
- Απόλυτο σφάλμα πρόβλεψης με ανώφλι $\rightarrow$ σφάλμα (9.891%)
- Οπτικοποίηση είτε ως εικόνα είτε ως ιστόγραμμα



(α') $\mathbf{AD} = |\mathbf{D}^L_{\mathbf{ground\_truth}} - \mathbf{D}^L_{\mathbf{predicted}}| \in [0, 60]px$

(β') $\mathbf{AD}_{\mathbf{threshold}} = |\mathbf{D}^L_{\mathbf{ground\_truth}} - \mathbf{D}^L_{\mathbf{predicted}}| \gtrless 3px$

Πως ψάχνουμε αντίστοιχα σημεία;

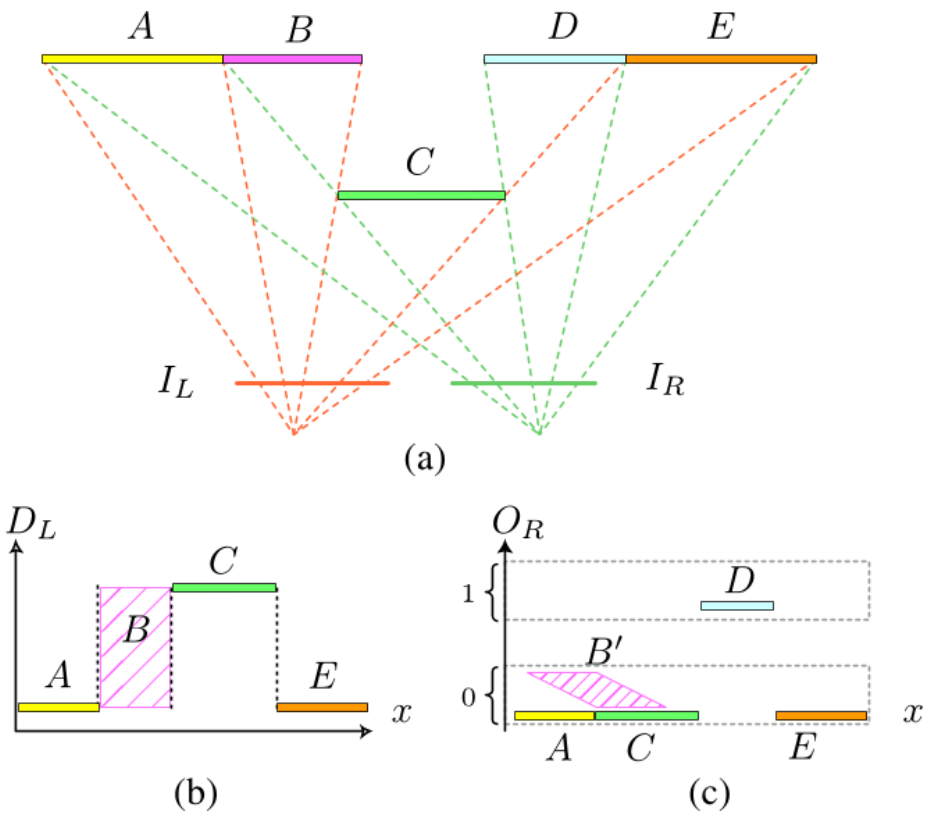
(i) **Ομοιότητα γειτονιάς**:

- Παρόμοιο σχήμα, υφή στις δύο προβολές
- small baseline (20cm)

(ii) **Στερεοσκοπικοί περιορισμοί**:

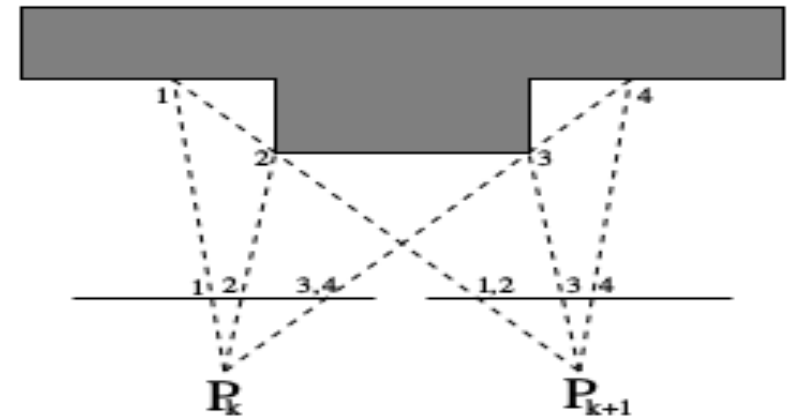
- Περιορισμός συνέχειας/ασυνέχειας
- Περιορισμός μοναδικότητας
- Συνέπεια διάταξης σημείων

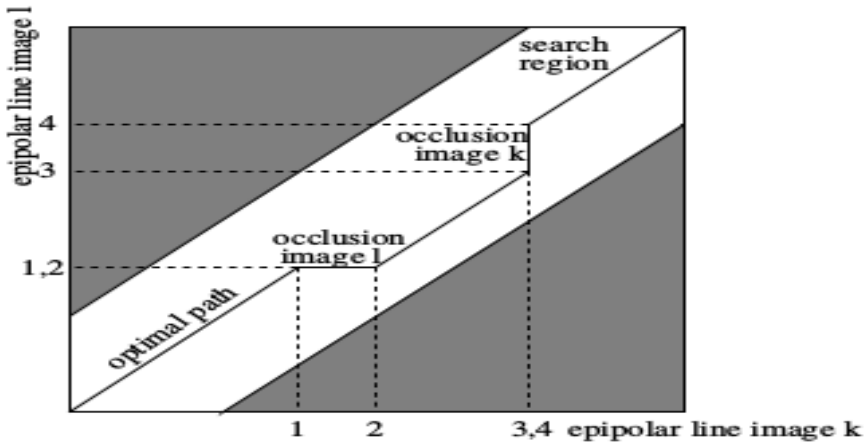
Ασυνεχείς επιφάνειες $\rightarrow$ ασυνέχειες παράλλαξης $+$ αποκρύψεις $\rightarrow$ πηγή προβλημάτων!



Περιορισμός μοναδικότητας

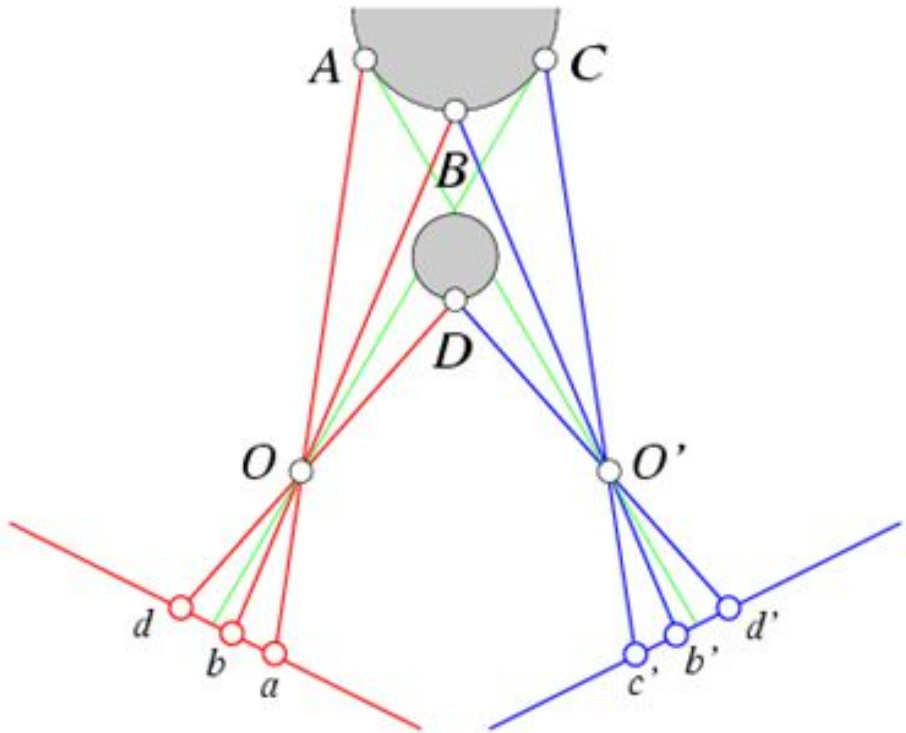
- Υπάρχει ‘1-1’ σχέση μεταξύ σημείων αριστερής/δεξιάς λήψης.
- Αίρεται όταν υπάρχουν αποκρύψεις





Συνέπεια διάταξης σημείων

- Αν για σημεία p_1, p_2 της μιας λήψης ισχύει $x_1 < x_2$ το ίδιο θα ισχύει και για τα «αντίστοιχα» σημεία τους
- Αίρεται όταν υπάρχουν αποκρύψεις



Επίλυση Προβλήματος

Χωρίζουμε το πρόβλημα σε 3 στάδια:

1. Δημιουργία πίνακα κόστους: $I^L, I^R \rightarrow C$
   $C(d, x, y)$: πόση ομοιότητα εμφανίζει η γειτονιά του $I^L(x, y)$ με αυτή του $I^R(x - d, y)$
2. $C_{init} \rightarrow C_{optimized} \xrightarrow{argmin_d C(d,x,y)} D_{init}$
3. $D_{init} \rightarrow D_{optimized}$

Πίνακας C: Απλές μέθοδοι vs Νευρωνικό Δίκτυο

- Άθροισμα απόλυτων διαφορών:

$$C(d,p) = -\sum_{q \in N_p} |I^L(q) - I^R(q-d)|$$

- Ομοιότητα συνημιτόνου:

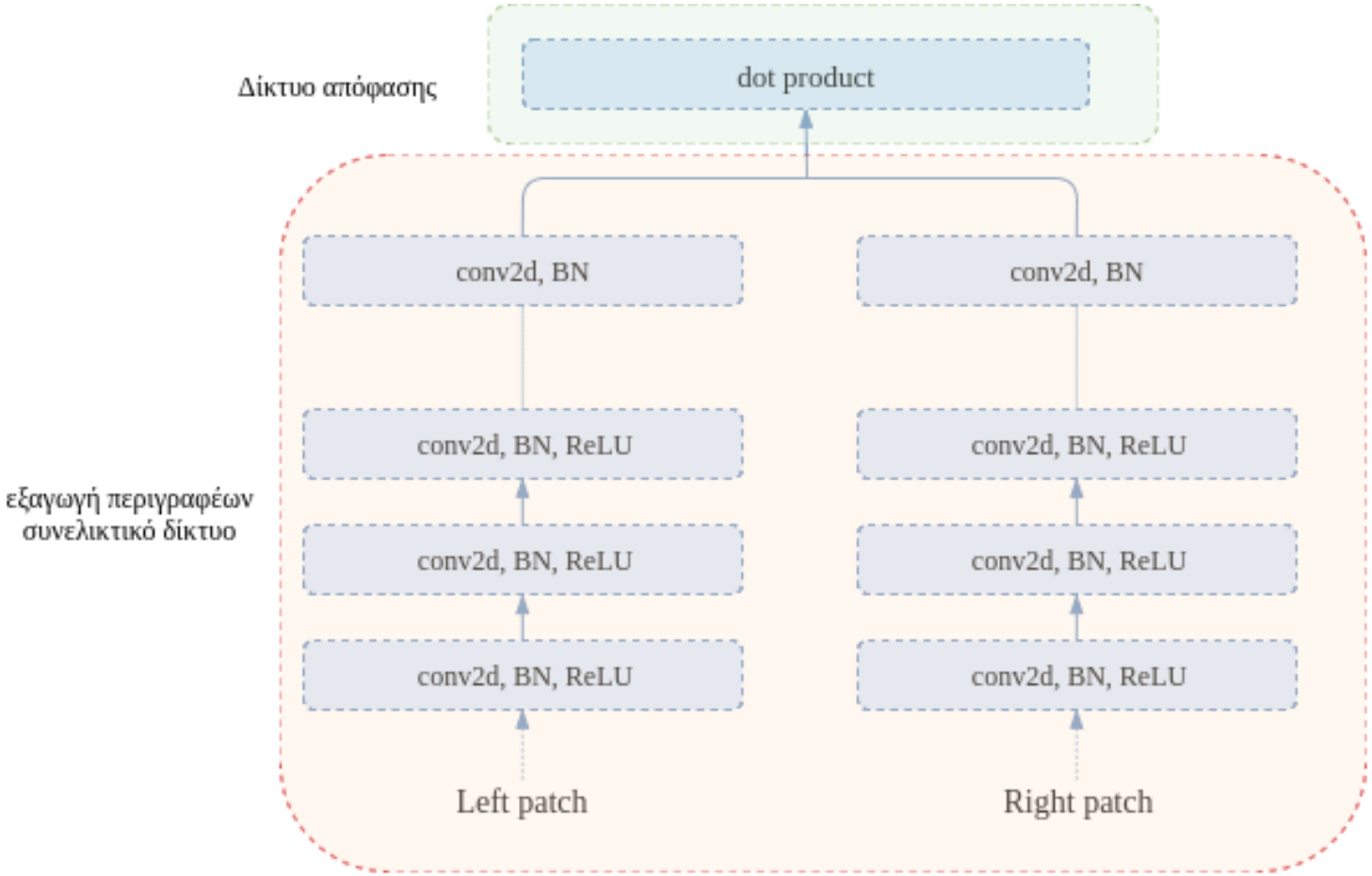
$$C(\mathbf{p},d) = \frac{\sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} I^L(\mathbf{q}) I^R(\mathbf{q}-\mathbf{d})}{\sqrt{\sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} I^L(\mathbf{q})^2 \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} I^R(\mathbf{q}-\mathbf{d})^2}}$$

- Απόσταση Hamming σε μετασχηματισμό Census

Νευρωνικό Δίκτυο

- Πλεονέκτημα:
  - Εξαγωγή βέλτιστου τοπικού περιγραφέα γειτονιάς από τα δεδομένα - ο σχεδιασμός του από προγραμματιστή θα ήταν αδύνατος
  - Βέλτιστος; Ενσωματώνει ποιοτικά χαρακτηριστικά (σχήμα, υφή κλπ) της γειτονιάς που μένουν αμετάβλητα στις δύο προβολές
- Μειονέκτημα:
  - Υπολογιστική πολυπλοκότητα: συνελικτικό νευρωνικό δίκτυο 8 κρυφών επιπέδων. Απαραίτητη η χρήση GPU. Ακόμα κι έτσι,
    Χρόνος εκτέλεσης $\approx 1 \frac{sec}{\text{εικόνα}}$

Αρχιτεκτονική νευρωνικού δικτύου



Είσοδοι συνελικτικού νευρωνικού δικτύου

Κατά την **εκπαίδευση**:

- **left patch:** γειτονιά N_p μεγέθους $[19 \times 19]$ γύρω από σημείο $I^L(\mathbf{p})$
- **right patch:** χωρίο μεγέθους $[(19 + \texttt{max\_disparity}) \times 19]$. Το χωρίο περιλαμβάνει τις γειτονιές N_p όλων την υποψήφιων «αντίστοιχων» σημείων $I^R(\mathbf{p} - \mathbf{d})$
- συνέλιξη χωρίς zero-padding $\rightarrow$ χωρικές διαστάσεις μειώνονται κατά 2 σε κάθε μπλοκ $(19 - (8 \cdot 2) = 1)$

(a) left patch

(b) right patch

Κατά την **εκτέλεση**:

- Ολόκληρες εικόνες $I^L, I^R \rightarrow$ με zero padding.

Έξοδοι συνελικτικού νευρωνικού δικτύου

Κατά την **εκπαίδευση**:

- $I^L_{desc}(\mathbf{p})$: τον descriptor του left patch
- τους $\texttt{max\_disparity} + 1$ descriptors των αντίστοιχων γειτονιών των σημείων $I^R(\mathbf{p} - \mathbf{d})\ \forall d \in [0, \ldots, \texttt{max\_disparity}]$
- $I^L_{desc}(\mathbf{p}) \in \mathbb{R}^{64}$

Κατά την **εκτέλεση**:

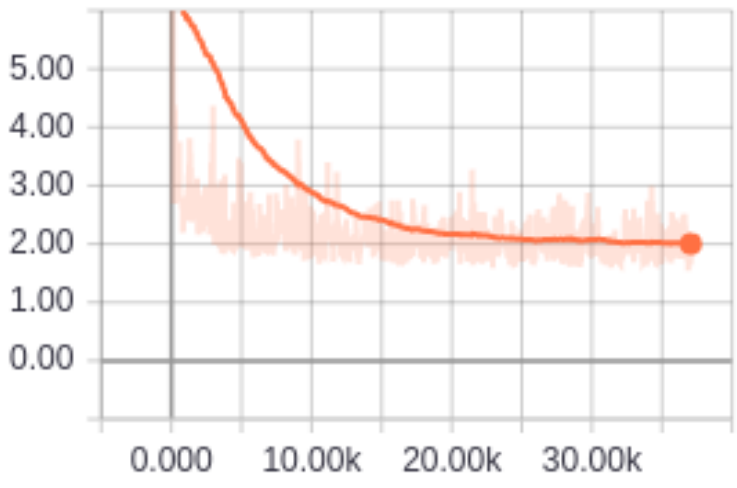
- I^L_{desc}, I^R_{desc}: descriptors κάθε θέσης αριστερής και δεξιάς εικόνες

Το δίκτυο απόφασης λειτουργεί όμοια σε εκπαίδευση/εκτέλεση:

- Συγκρίνουμε descriptors με dot product

Εκπαίδευση

- Σύγκριση $I^L_{desc}(\mathbf{p})$ και $I^R_{desc}(\mathbf{p} - \mathbf{d})\ \forall d \in [0, \ldots, \texttt{max\_disparity}]$. Αποθηκεύουμε τις τιμές στο διάνυσμα *similarity*.
- C_{data}: softmax cross entropy μεταξύ των διανυσμάτων *similarity* και *label* (δείχνει την σωστή τιμή παράλλαξης)
- $C_{reg} = \lambda \sum_i \Theta_i^2$, προσθέτουμε dropout(0.4) μετά από κάθε ReLU
- Optimizer: ADAM
- Εκπαίδευση σε $\approx 4 \times 10^6$ παραδείγματα
- batch size = 128



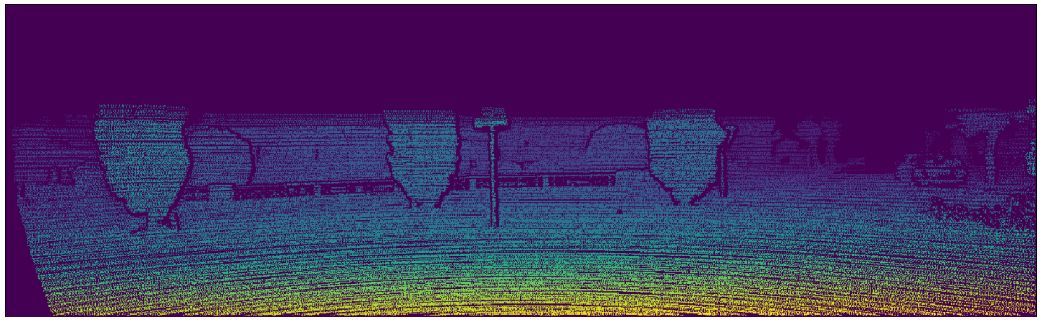
Εκπαίδευση vs εκτέλεση

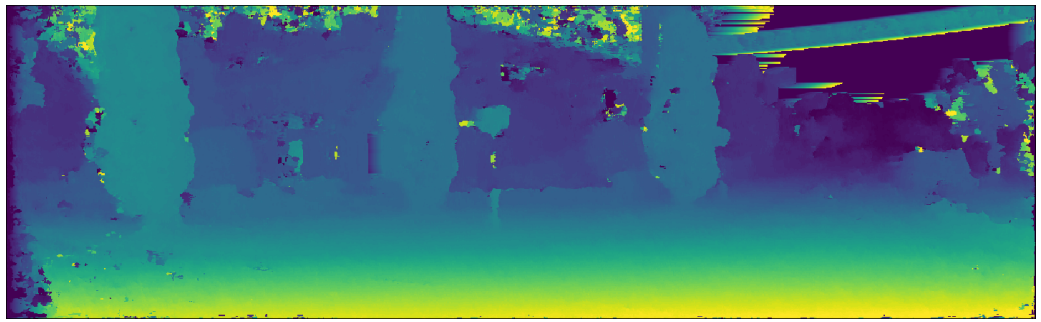
Γιατί διαφορετικές είσοδοι σε εκπαίδευση και εκτέλεση;

- Τρικ για την αύξηση του dataset. Κάθε εικόνα δημιουργεί $\approx 10^5$ παραδείγματα
- training set: 125 εικόνες $\rightarrow 15 \times 10^6$
- Τι χάνουμε; Περιορισμός στα layers που μπορούμε να χρησιμοποιήσουμε. Αποκλείονται layers που αλλοιώνουν χωρικές διαστάσεις (πχ max pool)

Αποτέλεσμα νευρωνικού δικτύου

- μέσο απόλυτο σφάλμα: $1.132px$
- ποσοστό σφάλματος: 5.846%
- μέθοδος AD-Census: 19.84%





Βήμα 2: $C_{init} \rightarrow C_{optimized}$

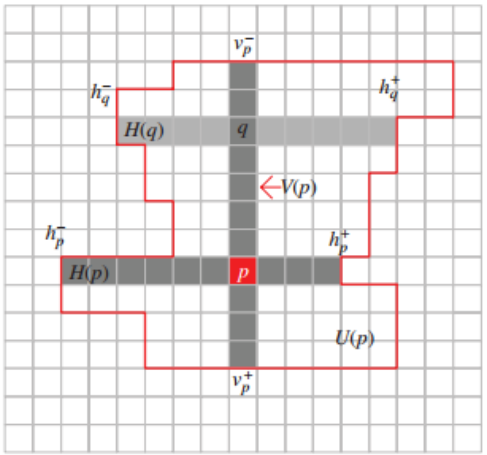
- συνεχείς περιοχές → συνεχής παράλλαξη
- Προσεκτική δημιουργία γειτονιών N_p με μέθοδο σταυρού (Mei et. al 2011)

Πήγαινε κάτω, πάνω, αριστερά, δεξιά όσο:

$|I(\mathbf{p}) - I(\mathbf{p}')| < \texttt{intensity\_threshold}$

$\|\mathbf{p} - \mathbf{p}'\| < \texttt{distance\_threshold}$

N_p : οριζόντιες ευθείες των σημείων της κάθετης ευθείας του σημείου p



Μέσος όρος κόστους σε γειτονιές

$\forall C(d, x, y)$:

- υπολόγισε την τομή των γειτονιών του $I^L(x, y)$ και του $I^R(x - d, y)$
- αποθήκευσε ως $C_{opt}(d, x, y)$ το μέσο όρο των C στην παραπάνω γειτονιά

Παρατηρήσεις:

- υποθέτει: δύο pixels στην ίδια γειτονιά θα έχουν συνήθως ίδιο ή κοντινό d
- εξομαλύνει αστάθειες στον αρχικό πίνακα κόστους
- μπορούμε να εφαρμόσουμε το βήμα επαναληπτικά n φορές

Semi-global matching (1)

$$E_C(D) = \sum_{\mathbf{p}} \Bigg(C(\mathbf{p}, D(\mathbf{p})) + \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} P_1 \cdot 1\{|D(\mathbf{p}) - D(\mathbf{q})| = 1\} + \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} P_2 \cdot 1\{|D(\mathbf{p}) - D(\mathbf{q})| > 1\} \Bigg)$$

Παρατηρήσεις:

- Heiko Hirschmüller - 2008
- εξομαλύνει τον πίνακα κόστους C λαμβάνοντας υπ' όψη το σύνολο των τιμών του
- οι «τιμωρίες» P_1 και P_2 μειώνονται αν οι φωτεινότητες γειτονικών pixel έχουν μεγάλη απόκλιση (πιθανή ακμή)
- αδύνατο να επιλυθεί καθολικά, οι πιθανές τιμές του D είναι $(m \times n)^{max\_disparity}$
- το αντιμετωπίζουμε με δυναμικό προγραμματισμό σε με μεμονωμένες κατευθύνσεις

Semi-global matching (2)

Επιλέγω κατεύθυνση r κι εφαρμόζω αναδρομική σχέση:

$$C_r(\mathbf{p}, d) = C(\mathbf{p}, d) - \min_k C_r(\mathbf{p} - \mathbf{r}, k) + \min\Big\{ C_r(\mathbf{p} - \mathbf{r}, d), C_r(\mathbf{p} - \mathbf{r}, d-1) + P_1, C_r(\mathbf{p} - \mathbf{r}, d+1) + P_1, \min_k C_r(\mathbf{p} - \mathbf{r}, k) + P_2 \Big\}$$

- Υπολογιστική πολυπλοκότητα $O(directions \cdot max\_disparity \cdot W \cdot H)$
- όσες περισσότερες κατευθύνσεις τόσο καλύτερο αποτέλεσμα, ο Hirschmüller προτείνει 16
- το εφαρμόζουμε σε 4 (πάνω, κάτω, αριστερά, δεξιά) για να γλιτώσουμε πολυπλοκότητα

Βήμα 3: $D_{init} \rightarrow D_{opt}$

Μετά τις βελτιώσεις στο πεδίο του C υπολογίζουμε τον χάρτη (εικόνα) παράλλαξης με μέθοδο winner takes it all:

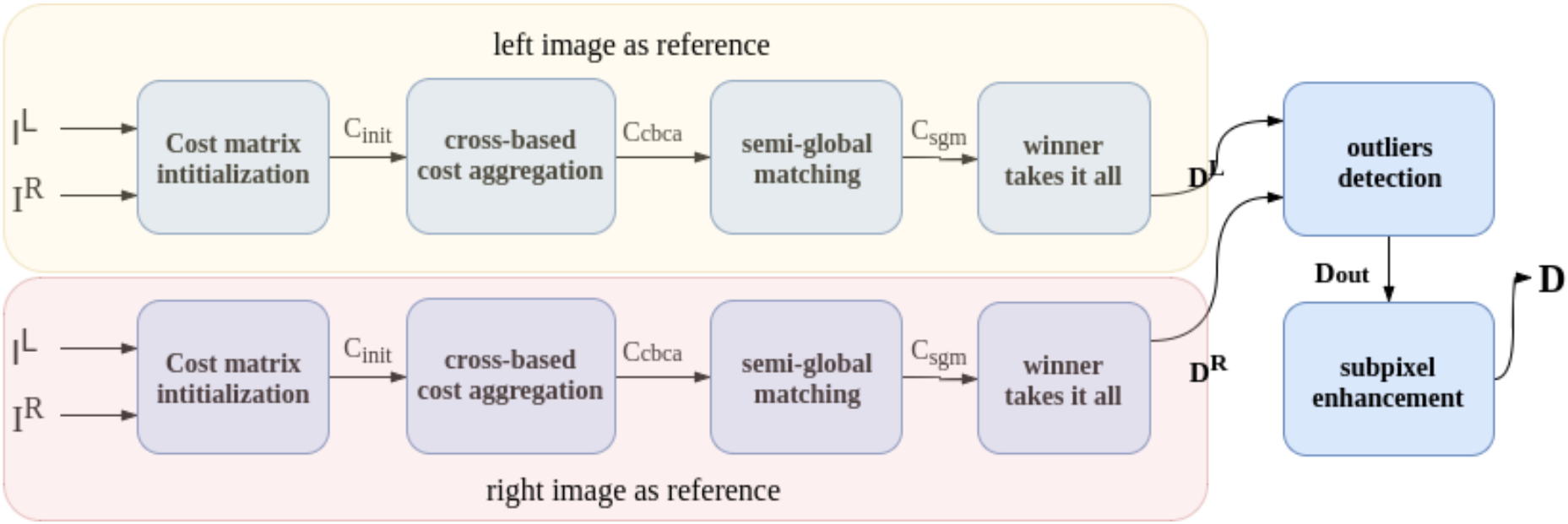
$$D = argmin_d C(d, x, y)$$

Υπολογίζουμε χάρτες D_L, D_R θεωρώντας εικόνα αναφοράς I_L, I_R αντίστοιχα και βασιζόμαστε στον περιορισμό μοναδικότητας:

- $|D^L(\mathbf{p}) - D^R(\mathbf{p} - \mathbf{d})| \leqslant 1$, τότε ορθή παράλλαξη και τέλος η αναζήτηση
- $|d - D^R(\mathbf{p} - \mathbf{d})| > 1 \quad \forall d : \{\mathbf{p} - \mathbf{d} \geqslant 0\}$, διατρέχω την επιπολική ευθεία κι ελέγχω ότι κανένα άλλο pixel δεν αντιστοιχίζεται με το pixel $\mathbf{p} \Rightarrow$ απόκρυψη
- αν αντίθετα υπάρχει $\Rightarrow$ αστοχία πρόβλεψης

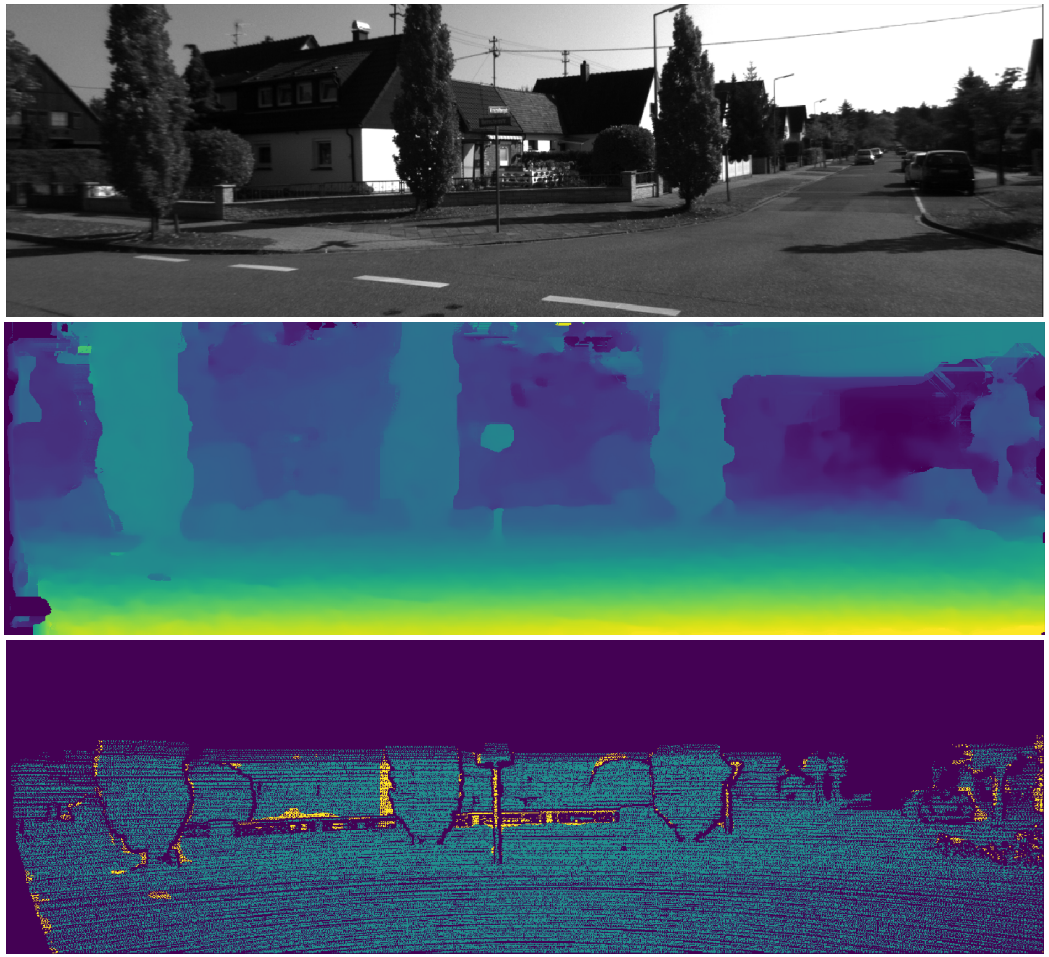
Επανυπολογίζουμε τις τιμές $D(p)$ με κατάλληλες μεθόδους παρεμβολής από τιμές γειτονικών pixel με σήμανση ορθής παράλλαξης

Συνολική μέθοδος



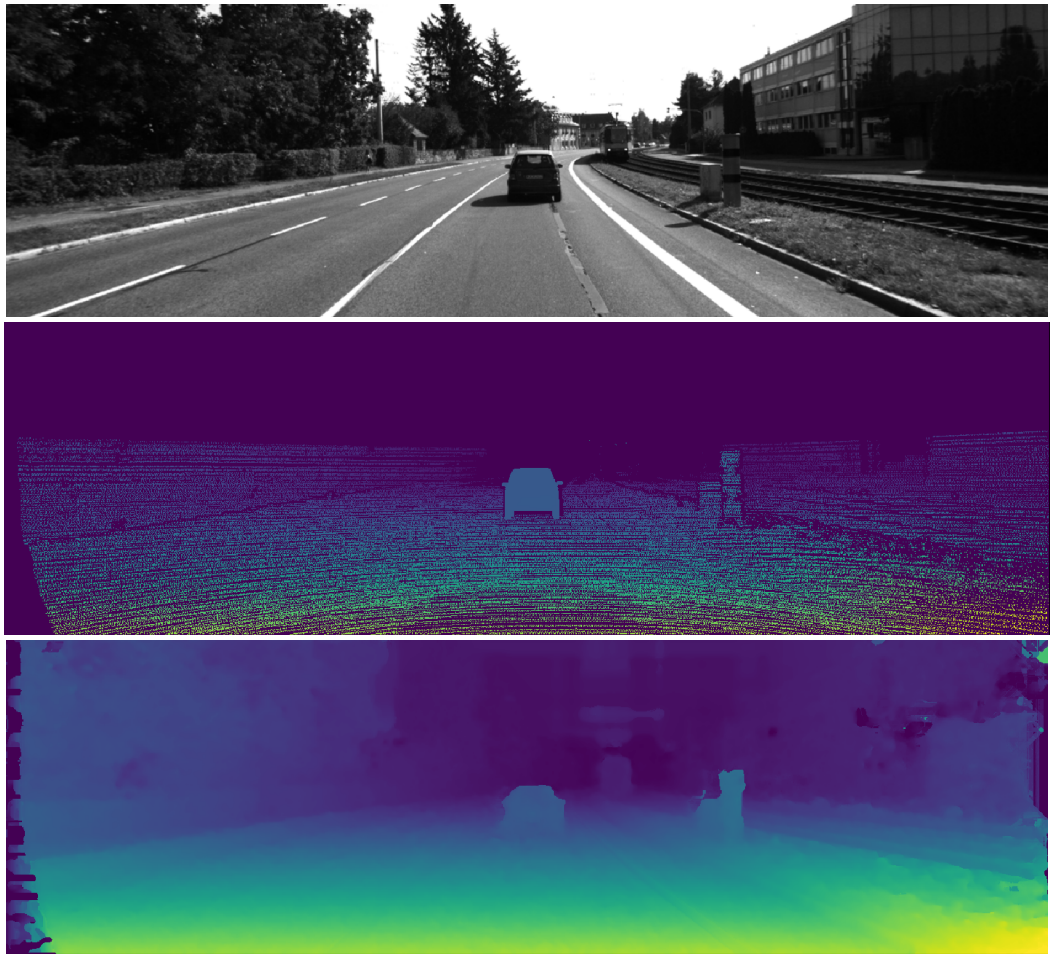
Ενδεικτικό παράδειγμα ολόκληρης μεθόδου

- μέσο απόλυτο σφάλμα: $0.62px$
- ποσοστό σφάλματος: 2.67%
- μέθοδος AD-Census: 5.422%



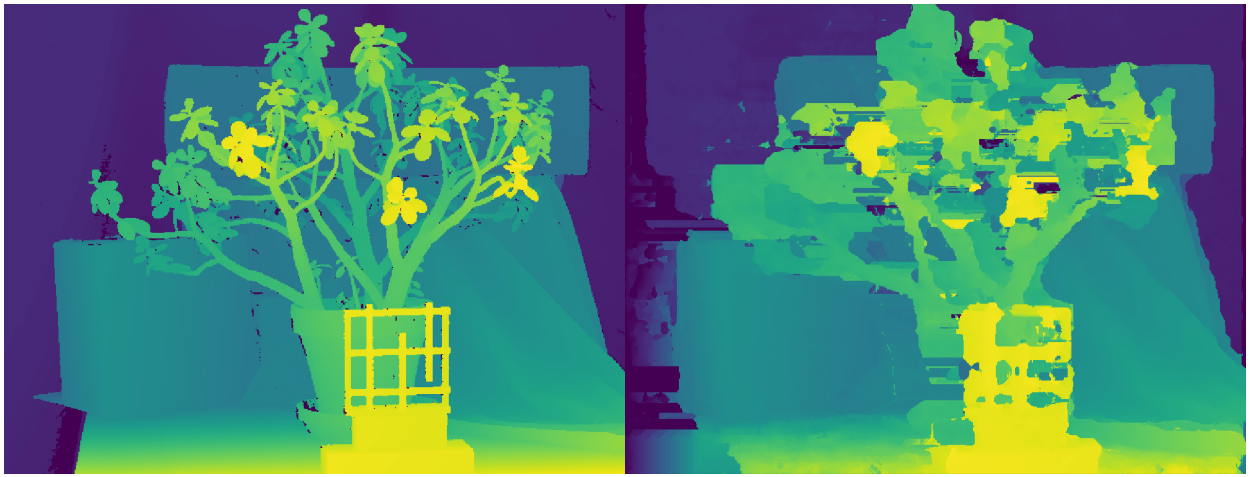
Από τα καλύτερα

- μέσο απόλυτο σφάλμα: $1.577px$
- ποσοστό σφάλματος: 1.248%

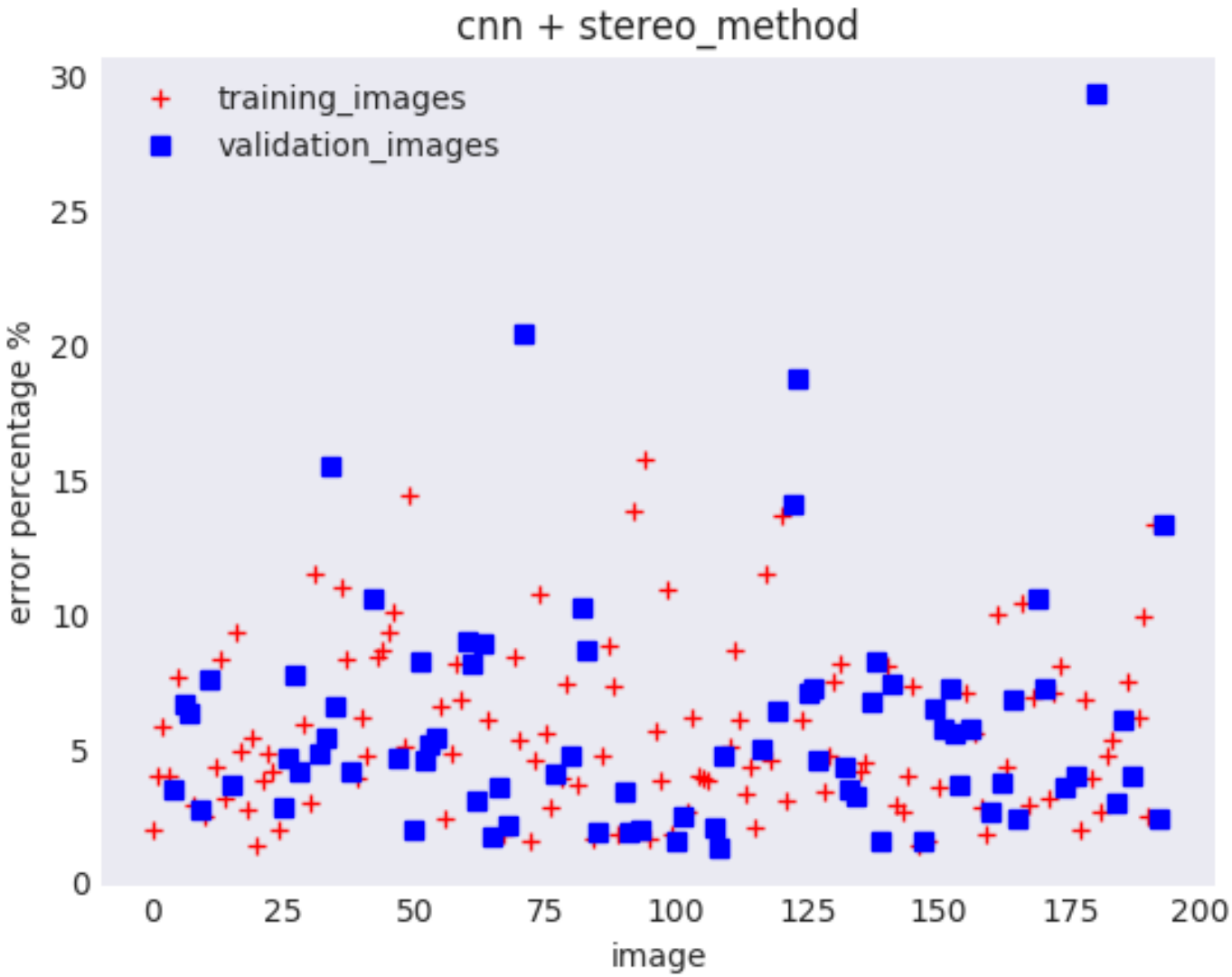


Από τα χειρότερα

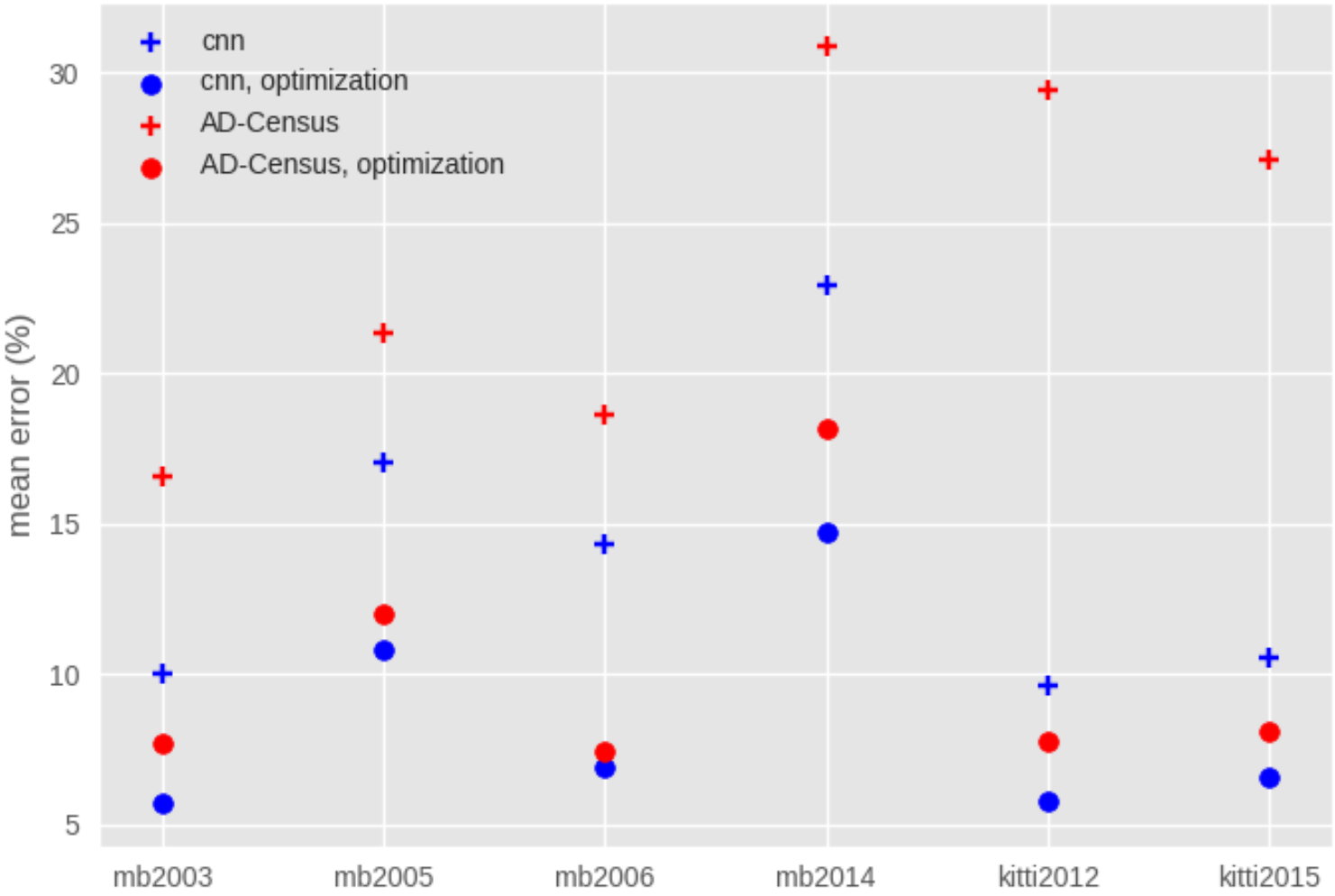
- μέσο απόλυτο σφάλμα: $9.98px$
- ποσοστό σφάλματος: 29.61%



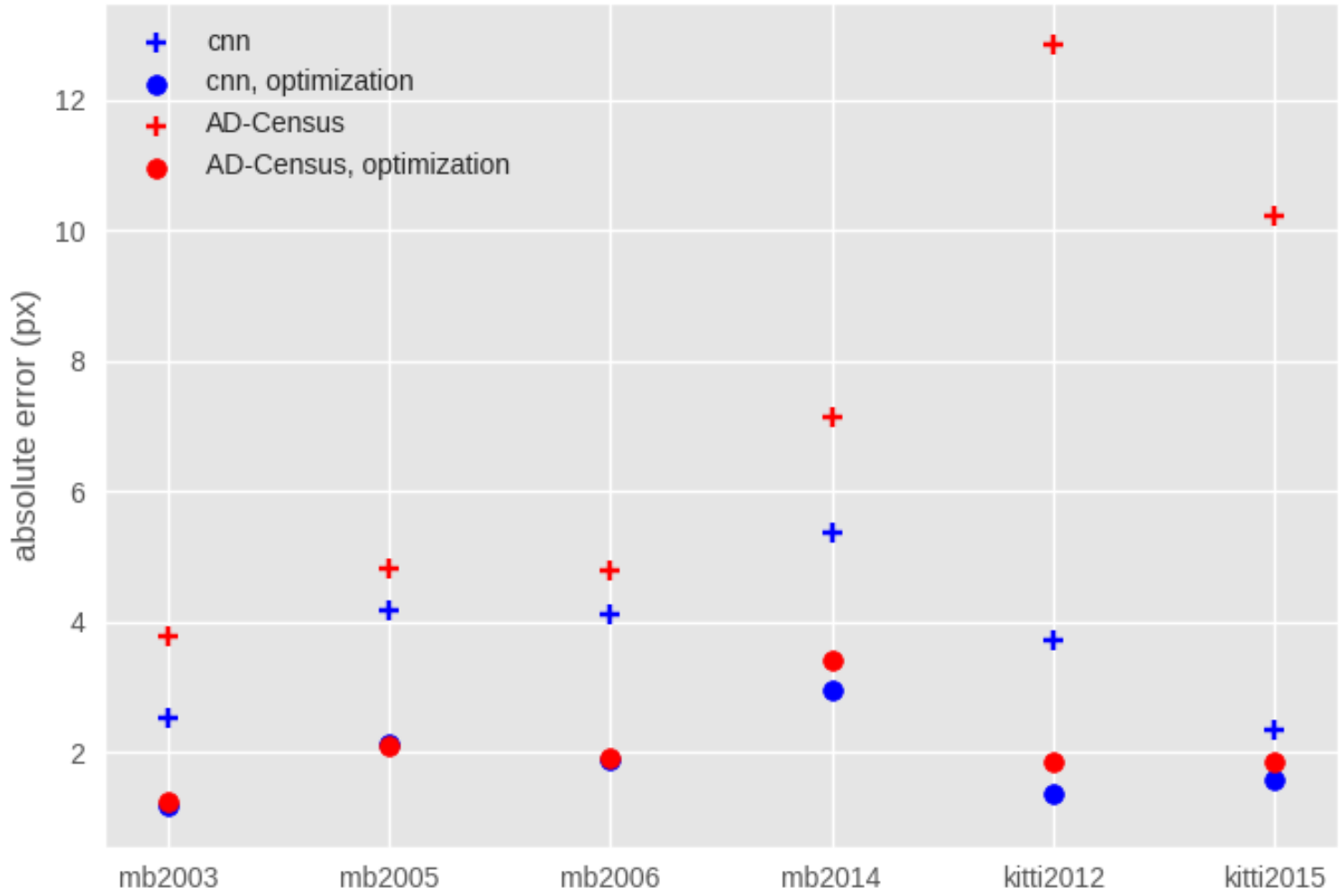
Αποτελέσματα - KITTI 2012



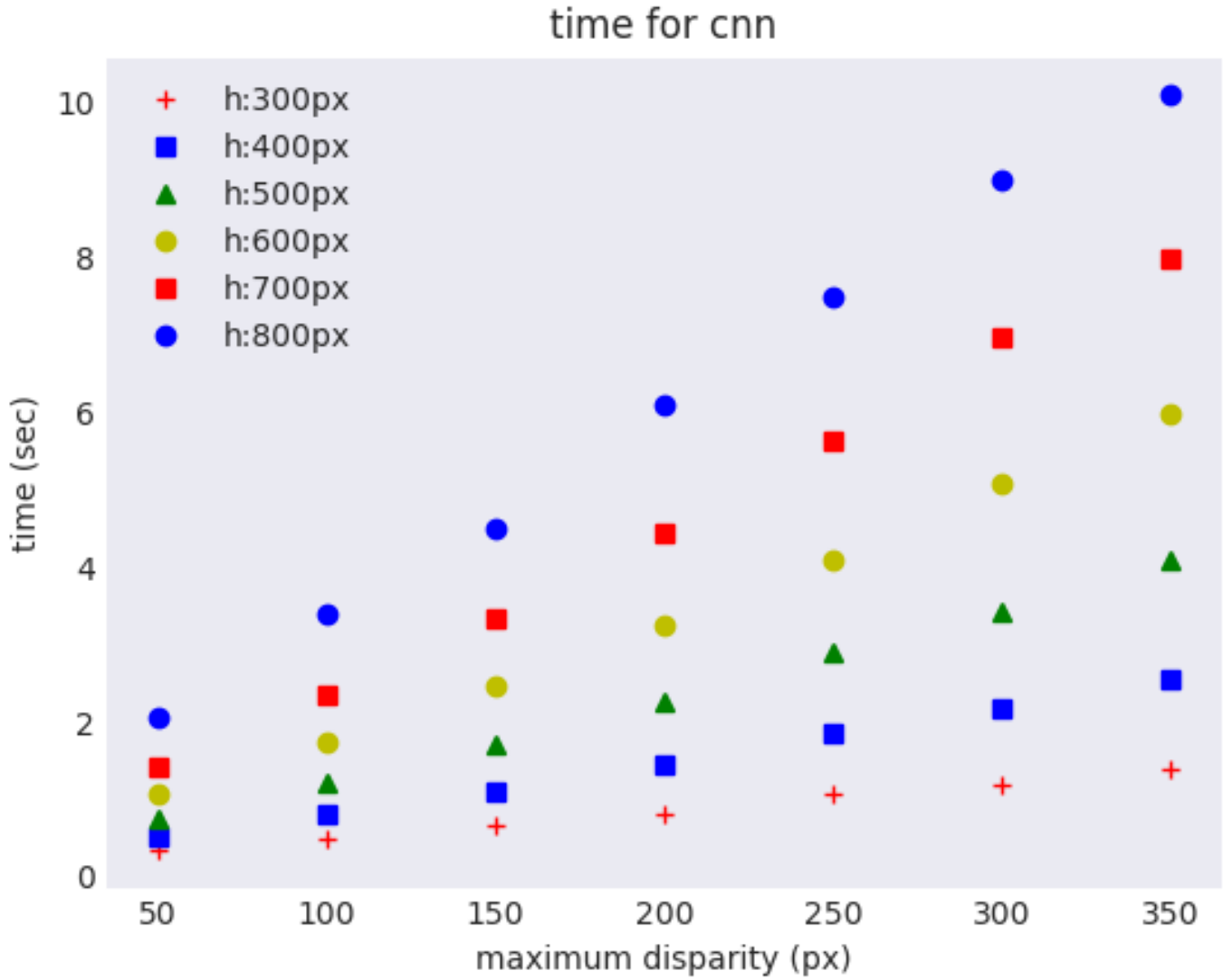
Αποτελέσματα - Απόλυτο σφάλμα με ανώφλι



Αποτελέσματα - Απόλυτο σφάλμα



Αποτελέσματα - Χρόνοι εκτέλεσης νευρωνικού



Συμπεράσματα

- Cnn βελτιώνουν εξαιρετικά την ακρίβεια της μεθοδολογίας
- Αρκετά καλή γενίκευση, ακόμα και σε στερεοσκοπικά ζεύγη τελείως διαφορετικής στατιστικής

Δυσκολίες:

- Υπολογιστική πολυπλοκότητα - απαραίτητη GPU - αδύνατο να τρέξει απευθείας σε hardware κινητού, απαραίτητη μεσολάβηση server
- Χρόνοι $\approx 1^{sec}/_{image}$ - χρειάζεται επιτάχυνση $25x$ για live video

Προτάσεις:

- Μοντελοποίηση βημάτων 2, 3 με μηχανική μάθηση (δυσκολία στην πράξη *argmin*)
- Εξερεύνηση τεχνικών unsupervised learning (πχ μέσω reprojection error)
- Βάθος από μια λήψη
- Μείωση χρόνου εκτέλεσης

Ενδεικτική Βιβλιογραφία

- Jure Zbontar and Yann LeCun (2016): "Stereo matching by training a convolutional neural network to compare image patches"
- Wenjie Luo, Alexander G. Schwing and Raquel Urtasun (2016): "Efficient Deep Learn- ing for Stereo Matching"
- Xing Mei et al (2011): "On building an accurate stereo matching system on graphics hardware"
- Heiko Hirschmuller (2008): "Stereo processing by semiglobal matching and mutual in- formation"

Τέλος

Ευχαριστώ πολύ για την προσοχή σας!
Ερωτήσεις;