



האוניברסיטה הפתוחה

המחלקה למתמטיקה ולמדעי המחשב

החטיבה למדעי המחשב

# פרויקט מתקדם במדעי המחשב

## מימוש שיטת AD Census להתאמת סטראו

מנחים: ד"ר עזריה הכהן, ד"ר מיריי אביגל

מגיש:

שחר שר שלום

ת.ז. 300394160

סמסטר ב' 2018

**דוח פרויקט זה הוגש כחלק מהדרישות לקבלת תואר "מוסמך (M.Sc.) במדעי המחשב"**

**באוניברסיטה הפתוחה**

**מאי 2018**

## 1. תוכן עניינים

2	.....	תוכן עניינים	1
3	.....	רשימת איורים	2
5	.....	מבוא	3
5	.....	מודל מצלמה	4
6	.....	גיאומטריה של מודל מצלמה	4.1
6	.....	מציאת עומק משני מודלי מצלמה	4.2
8	.....	אלגוריתם AD CENSUS	5
8	.....	שלב א' AD – CENSUS COST FUNCTION	5.1
15	.....	שלב ב' CROSS BASED COST AGGREGATION	5.2
21	.....	שלב ג' SEMI GLOBAL MATCHING	5.3
26	.....	שלב ד' MULTISTEP DISPARITY REFINEMENT	5.4
29	.....	מימוש אלגוריתם AD-CENSUS	6
29	.....	מימוש האלגוריתם ב C++   MATLAB	6.1
31	.....	סיבוכיות זיכרון	6.2
32	.....	מבנה מודולי תוכנה	6.3
33	.....	ממשק משתמש	6.4
34	.....	תוצאות ומסקנות	7
34	.....	מבחני הערכת האלגוריתם	7.1
35	.....	תוצאות	7.2
40	.....	סיכום	8
41	.....	ביבליוגרפיה	9

## 2. רשימת איורים

- איור 1 - דוגמה לשימוש באלגוריתם להתאמת סטראו [3]..... 5
- איור 2 - מודל מצלמה [6]..... 6
- איור 3 - מיפוי נקודה בעולם באמצעות שני מודולי מצלמה [4]..... 7
- איור 4 - דוגמה לשתי תמונות מיושרות אפיפולרית..... 8
- איור 5 - דוגמה לתמונות סטראו..... 10
- איור 6 - ימין - תמונת העומקים שהתקבלה משימוש בפונקציית עלות AD באמצעות תמונות הסטראו מאיור 5, שמאל - Ground Truth עבור התמונות מאיור 5..... 11
- איור 7 - דוגמה להפעלת Census Transform על גבי חלון של פיקסלים..... 12
- איור 8 - דוגמה לשימוש בפונקציית עלות Census המתקבלת על ידי שני חלונות של פיקסלים..... 12
- איור 9 - ימין - מפת העומקים שהתקבלה משימוש בפונקציית עלות Census, אמצע - תמונת העומקים שהתקבלה משימוש בפונקציית העלות AD, שמאל - Ground Truth..... 13
- איור 10 - דוגמה לפונקציה הנרמול  $\rho(c, \lambda)$ ..... 14
- איור 11 - ימין עליון - מפת העומקים שהתקבלה משימוש בפונקציית העלות AD-Census, שמאל עליון - Ground Truth, ימין תחתון - מפת העומקים שהתקבלה משימוש בפונקציית עלות Census, שמאל תחתון - תמונת העומקים שהתקבלה משימוש בפונקציית העלות AD..... 14
- איור 12 - מבנה השלד הראשוני (קווי המתאר של הסגמנט)..... 16
- איור 13 - דוגמה לחישוב אזור הסכימה מתוך השלד הראשוני בעזרת סביבה של פיקסלים..... 18
- איור 14 - מספר דוגמאות לשלד הראשוני (קווי המתאר של הסגמנט) המתקבל עבור סביבה של פיקסלים..... 18
- איור 15 - מספר דוגמאות לאזור הסכימה המתקבל עבור סביבה של פיקסלים..... 18
- איור 16 - חישוב שטח בתחום מלבני באמצעות Integral Image..... 20
- איור 17 - דוגמה למתודולוגיה מיצוע אזור הסכימה..... 20
- איור 18 - ימין - מפת העומקים שהתקבלה משימוש בפונקציית העלות משלב ב' Cross Based Cost Aggregation, אמצע - מפת העומקים שהתקבלה משימוש בפונקציית העלות משלב א' AD-Census, שמאל - Ground Truth..... 21

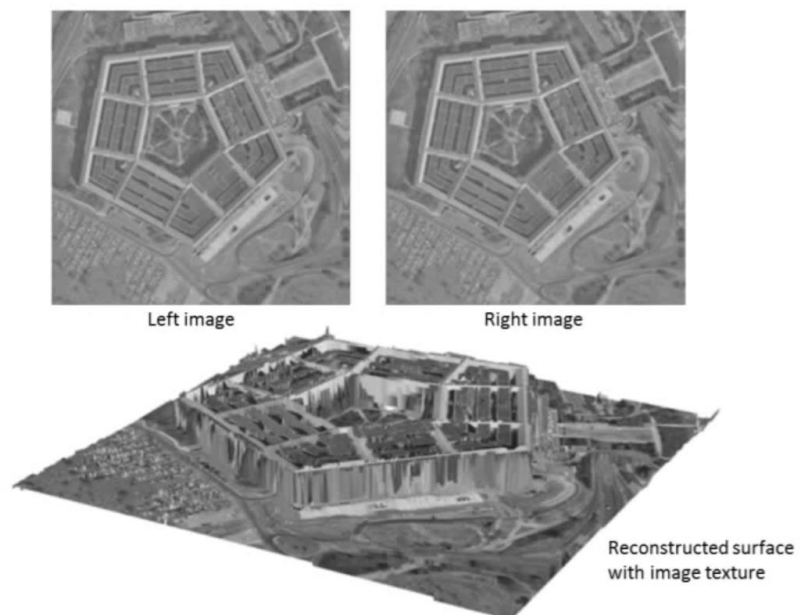
- איור 19 – פונקציית העלות SGM המתקבלת משימוש ב 16 קווי סריקה מכיוונים שונים [12].....23
- איור 20 - תמונות העומקים שהתקבלו משימוש באלגוריתם קו סריקה בכיוונים שונים.....25
- איור 21 – ימין - תמונת העומקים שהתקבלה משימוש באלגוריתם Semi Global Matching, אמצע -  
 מפת העומקים שהתקבלה משימוש ב Cross based cost aggregation, שמאל - Ground Truth ...26
- איור 22 – כיווני האינטרפולציה באלגוריתם Proper Interpolation ..28
- איור 23 – ימין - תמונת העומקים שהתקבלה משימוש בשלב ד' של האלגוריתם, אמצע - תמונת  
 העומקים המתקבלת לאחר הפעלת אלגוריתם קו סריקה, שמאל - Ground Truth .....29
- איור 24 – תרשים עיבוד מקבילי של האלגוריתם.....31
- איור 25 - מבנה עיצוב התוכנה.....33
- איור 26 – ממשק משתמש (GUI) להפעלת האלגוריתם.....33
- איור 27 – פירוט החלקים השונים מאיור 26.....34
- איור 28 – ביצועי המעבדים ונצילותם במהלך הרצת האלגוריתם.....37

### 3. מברא

מצלמות מודרניות מצלמות בדו ממד. על ידי כלים מתמטיים ניתן לשלב מספר תמונות דו ממדיות בכדי ליצור תמונה תלת ממדית של הסצנה המקורית. בפרויקט זה נסקור מספר אלגוריתמים שונים לבניית מודל תלת ממדי מתוך אוסף תמונות דו ממדיות, ונשווה בניהם על ידי מדדים מקובלים להשוואת אלגוריתמים.

בני אדם מודדים מרחק בעזרת שתי תמונות, תמונה מעין ימין ותמונה מעין שמאל. שתי העיניים מרוחקות זו מזו ולכן מתקבלות שתי תמונות דומות אך בעלות שוני הנובע מזווית ההסתכלות בניהן (תופעת הפרלקסה). המוח מאחד בין התמונות ובכך בני האדם מודדים מרחק. אוסף תמונות המתקבלות מאותה הסצנה מזוויות שונות נקראות **תמונות סטראו**. תהליך התאמת תמונות סטראו לתמונה תלת ממדית נקרא **התאמת סטראו** (Stereo Matching).

באיור 1 ניתן לראות דוגמה לתוצאת אלגוריתם להתאמת סטראו שבאמצעותה ניתן לבנות מודל תלת ממדי (תמונה תחתונה) מתוך שתי תמונות דו ממדיות.



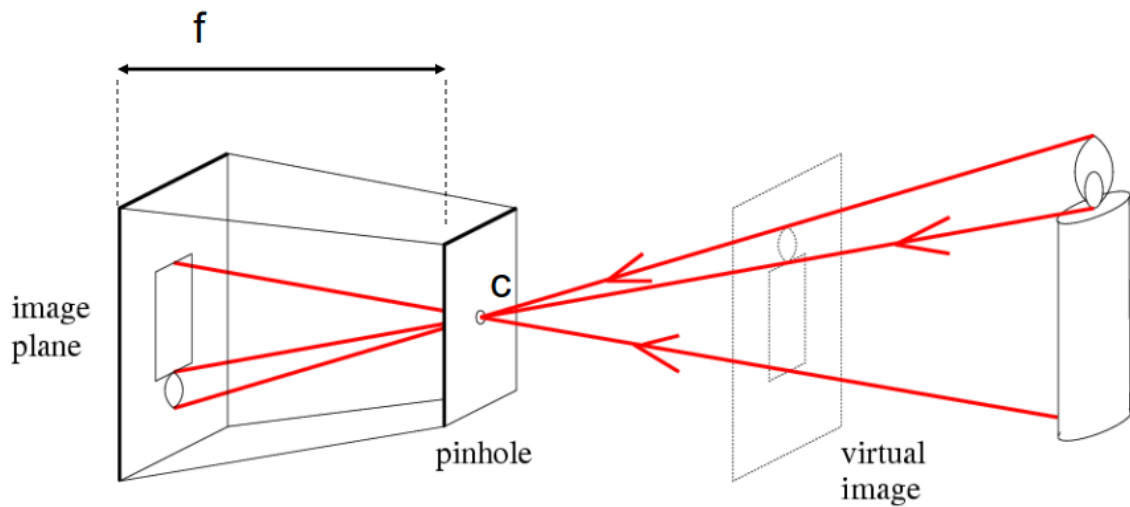
איור 1 - דוגמה לשימוש באלגוריתם להתאמת סטראו [3]

### 4. מודל מצלמה

נתאר מודל מתמטי של מצלמה, הנקרא **מודל מצלמה** (Camera Model) ומתאר את הקשר בין נקודות בעולם התלת ממדי לנקודות בעולם הדו ממדי [1] [2].

#### 4.1 גיאומטריה של מודל מצלמה

נניח קיומה של מערכת הכוללת שני מישורים מקבילים, כאשר במישור הראשון יש חור (Pinhole) שדרכו עוברות קרני אור שנוצרו על ידי אובייקט בעולם התלת ממדי. קרניים אלו משתקפות ויוצרות תמונה הפוכה על גבי המישור השני, הנקרא **מישור התמונה** (Image Plane). מישור התמונה מתקבל בצורתו ההפוכה ב**מישור התמונה הוירטואלי** (Virtual Image Plane), ראה איור 2. המצלמה נמצאת בין החור שדרכו עוברות קרני האור עד למישור התמונה. מודל בסיסי זה נקרא מודל מצלמה.



איור 2 - מודל מצלמה [6]

על גבי מישור התמונה יוטלו נקודות העולם התלת ממדי לעולם הדו ממדי. נסמן את אוסף הנקודות בעולם התלת ממדי ב  $\mathcal{R}^3$ , ונסמן את אוסף הנקודות במישור התמונה בעולם הדו ממדי ב  $\mathcal{R}^2$ . המרחק בין מרכז החור למישור התמונה נקרא **מרחק המוקד** (Focal Length) ונסמן מרחק זה ב  $f$ . תהי  $P_{world} = (X, Y, Z)^T \in \mathcal{R}^3$  נקודה בעולם התלת ממדי, שבה הפרמטרים  $X, Y, Z$  מתארים את מיקום הנקודה בעולם והפרמטר  $Z$  מתאר את עומק הנקודה בעולם. מִמְפָּה לנקודה במישור התמונה על ידי הנוסחה הבאה

$$p_{image} = \left( \frac{fX}{Z}, \frac{fY}{Z} \right) \in \mathcal{R}^2$$

#### 4.2 מציאת עומק משני מודלי מצלמה

נניח כי נתונות שתי תמונות, תמונה ימין ותמונה שמאל, המתקבלות על ידי שתי מצלמות בעלות מרחק מוקד זהה הממפות עולם תלת ממדי משותף, ראה באיור 3 דוגמה למקרה זה. נתבונן בנקודה בעולם  $P_{world}$  המשותפת לשתי התמונות ונמפה את הנקודה  $P_{world}$  לנקודה  $(x_l, y_l)^T \in \mathcal{R}^2$

במישור התמונה השמאלית, ולנקודה  $(x_r, y_r)^T \in \mathcal{R}^2$  במישור התמונה הימנית. המרחק בין מישור התמונה השמאלית למישור התמונה הימנית נקרא **קו בסיס** (Baseline) ונסמן אותו ב  $b$ . אם הציר האופקי של מישור התמונה השמאלית מתלכד עם הציר האופקי של מישור התמונה הימנית, אזי נאמר כי שתי התמונות מוגדרות **כמישורות אפיפולריות** (Epipolar Rectify) [3]. בפרויקט זה נניח כי תמונות הסטראו מישורות אפיפולריות.

על סמך מיפוי  $P_{world}$  ל  $(x_l, y_l)^T$  במישור התמונה השמאלית ול-  $(x_r, y_r)^T$  במישור התמונה הימנית נקבל

$$x_l = f \frac{X}{Z}$$

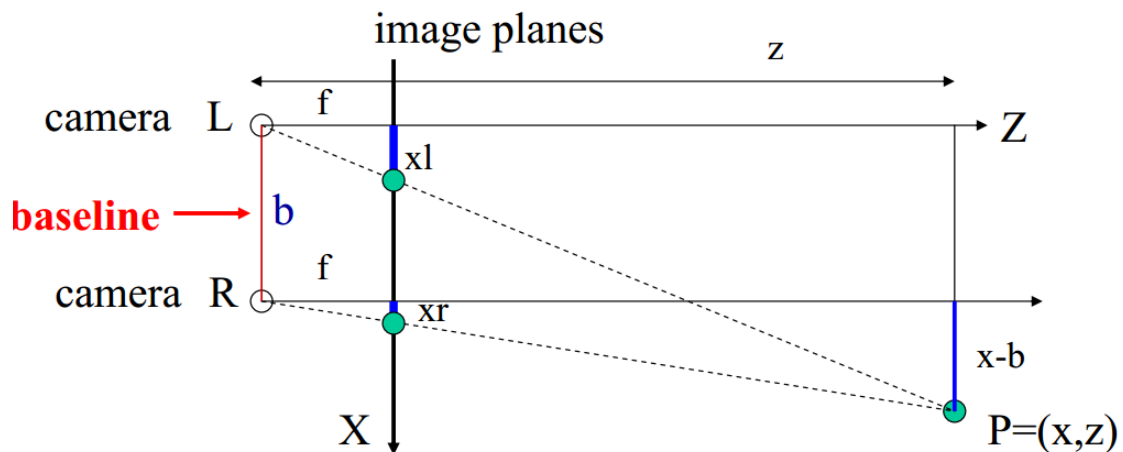
$$x_r = f \frac{X - b}{Z}$$

משוואה 1

$$x_l - x_r = \text{disparity} = f \frac{b}{Z}$$

משוואה 2

$$Z = \frac{fb}{\text{disparity}}$$



איור 3 - מיפוי נקודה בעולם באמצעות שני מודולי מצלמה [4]

ההפרש בין הנקודה במישור התמונה השמאלית לנקודה במישור התמונה הימנית נקרא **Disparity**. מכיוון ש  $f$  ו  $b$  הם קבועים במשוואה 2, בעיית מציאת העומק  $Z$  שקולה למציאת ה **Disparity**.

באיור 4 ניתן לראות דוגמה לשתי תמונות מיושרות אפיפולרית בהן הציר האופקי של שתי התמונות מתלכד. ה Disparity עבור הנקודה האדומה בתמונה מוגדר כהפרש הקואורדינטות בין  $x_l$  ל  $x_r$ .



איור 4 - דוגמה לשתי תמונות מיושרות אפיפולרית

## 5. אלגוריתם AD Census

להלן תוצג שיטה להתאמת סטראו [5] המתאימה למערכות זמן אמת. האלגוריתם ניתן למימוש על גבי פלטפורמה לחישוב מקבילי, למשל כרטיס גרפי מרובה ליבות [6]. השיטה כוללת את ארבעת השלבים הבאים.

### 5.1 שלב א' AD – Census Cost function

בהינתן שתי תמונות: תמונה ימין ותמונה שמאל, מציאת עומקו של פיקסל שקולה למציאת ה Disparity בין שתי התמונות. נסמן את **ממדי התמונה** (Image Dimensions) ב  $M$  ו  $N$ , ונסמן **פיקסל** (Pixel) בתמונה השמאלית על ידי  $p$ .

מציאת ה Disparity עבור כל פיקסל בתמונה השמאלית, תבוצע על ידי מציאת הפיקסל התואם בתמונה הימנית, התאמה תבוצע על ידי חיפוש פיקסל תואם ב**טווח חיפוש** של פיקסלים שיסומן ב  $nDisp$  (Number of Disparities). הפיקסל התואם ימצא בהזזה של עד  $nDisp$  פיקסלים. נסמן את ערך ה Disparity ב  $d$ .

$$d \in [0, nDisp)$$

הפיקסל התואם לפיקסל  $p$  בתמונה הימנית ימצא בהזזה של  $d$  פיקסלים ויסומן ב  $pd$

$$pd = p \pm d$$



**ערך הפיקסל בתמונה השמאלית** עבורו תבוצע התאמה לפיקסל בתמונה הימנית יסומן ב  $I^L(p)$ ,  
**וערך הפיקסל בתמונה הימנית** התואם ל  $I^L(p)$  יקרא  $I^R(pd)$ .

על מנת לבדוק האם  $I^L(p)$  תואם ל  $I^R(pd)$ , נחשב באמצעות פונקציה המקבלת את  $p$  ו- $pd$  ושתי התמונות את **מידת ההתאמה**. פונקציות המשמשות לחישוב מידת ההתאמה נקראות **פונקציות עלות** (Cost Function), פונקציות אלו נחקרו רבות וניתן למצוא פונקציות כאלו בספרות.

נגדיר את פונקציית העלות

$$C(p, pd) \in \mathbb{R}$$

לכל פיקסל יבחר Disparity  $d$  בעל פונקציית עלות מינימלית

משוואה 3

$$d = \operatorname{argmin}_{\hat{a}}(C(p, pd(\hat{d})))$$

מציאת ה Disparity שקולה למינימיזציה על פונקציית העלות, ובחירת Disparity בעל פונקציית עלות מינימלית. נציג את פונקציות העלות AD, AD-Census ו Census.

### 5.1.1 Absolute Differences

נציג פונקציית עלות נפוצה בספרות, פונקציה זו סוכמת את הפרשי הפיקסלים בערכם המוחלט, נקראת **Absolute Differences (AD)** ונסמן אותה ב  $C_{AD}(p, pd)$ .

$$C_{AD}(p, pd) = \frac{1}{3} \sum_{i=R,G,B} |I_i^L(p) - I_i^R(pd)|$$

יתרונותיה של פונקציית העלות AD היא יעילות החישוב, מכיוון שמתבצעת החסרה של פיקסל בודד ולא סביבה של פיקסלים ניתן לחשב את פונקציית העלות ביעילות. מכיוון שפיקסלי המקור לפונקציית העלות אינם תלויים אחד בשני, ניתן לחשב את פונקציית העלות במקביל. מאידך, פונקציית העלות יכולה להיכשל לפרקים מכיוון שאינה מחושבת באמצעות סביבה של פיקסלים, בעיקר כאשר מדובר בתמונה בעלת סביבה עם צבע דומה - **סביבה הומוגנית**.

באיור 6 מוצגת תמונה בגווני אפור. גווני האפור השונים מעידים על עומקים שונים, ככל שערך הגוון בהיר יותר, עומקו של הפיקסל קטן יותר, כלומר הוא קרוב יותר למצלמה. כמו כן, ככל שערך הגוון כהה יותר, עומקו של הפיקסל גדול יותר, כלומר הוא רחוק יותר מהמצלמה. נקרא לתמונה זו **תמונת עומקים** (Depth Image).



איור 5 – דוגמה לתמונות סטראו, ימין - תמונה המתקבלת מעין ימין הנכנסת לאלגוריתם להתאמת סטראו, שמאל - תמונה המתקבלת מעין שמאל הנכנסת לאלגוריתם להתאמת סטראו

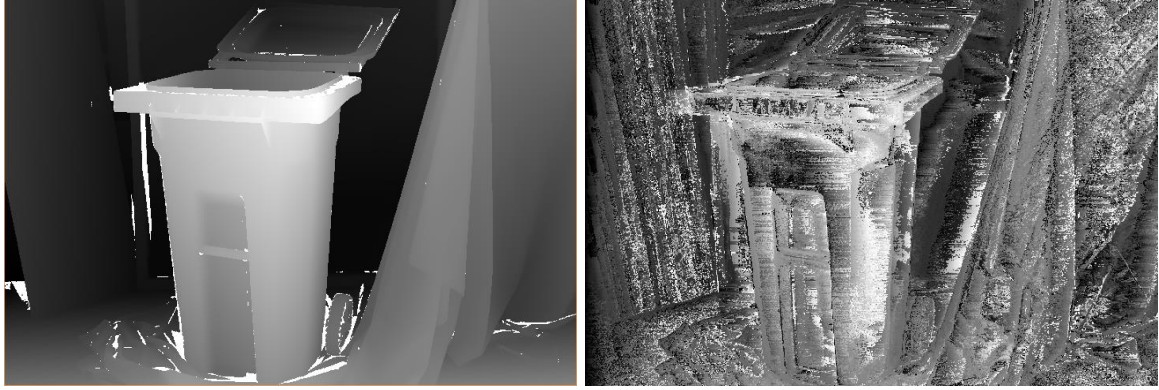
על מנת לאמוד את תוצאות התאמת הסטראו, נשתמש בתמונות סטראו הכוללות תוצאות **Ground Truth**. תמונות סטראו אלו כוללות תמונת עומקים מדויקת עם ערכי Disparity עבור כל פיקסל. באמצעות ה **Ground Truth** ניתן להעריך את מידת ההצלחה של אלגוריתם להתאמת סטראו. אחד מהמדדים הנפוצים להערכת אלגוריתם להתאמת סטראו הוא אחוז הפיקסלים השגויים בין תוצאות התאמת הסטראו ל **Ground Truth**. נקרא למדד זה בשם **PER – Pixel Error Rate**.

$$f(i, j) = \begin{cases} 1, & \text{if } (D_{AD-Census}(i, j) - D_{Ground Truth}(i, j)) \geq 1 \\ 0, & \text{Otherwise} \end{cases}$$

$$PER = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j)$$

נקרא ל  $D_{Ground Truth}(i, j)$  ערך ה Disparity המתקבל מתמונת ה **Ground Truth** עבור פיקסל  $(i, j)$ , נקרא ל  $D_{AD-Census}(i, j)$  ערך ה Disparity המתקבל באמצעות אלגוריתם להתאמת סטראו AD-Census עבור פיקסל  $(i, j)$ .

באיור 6 ניתן לראות את תמונת העומקים שהתקבלה מהפעלת משוואה 3 על פונקציית העלות  $C_{AD}(p, pd)$  שחושבה מתמונות הסטראו מאיור 5, בבדיקה מול **Ground Truth** התקבלה תוצאה  $PER = 74\%$ .



איור 6 - ימין - תמונת העומקים שהתקבלה משימוש בפונקציית עלות AD באמצעות תמונת הסטראו מאיור 5, שמאל - Ground Truth עבור התמונת מאיור 5

### 5.1.2. Census

נציג פונקציית עלות נוספת המבוססת על יחס בין פיקסלים שכנים בחלון פיקסלים בתמונה הימנית ויחס בין פיקסלים שכנים בחלון פיקסלים בתמונה השמאלית. נקרא לפונקציית עלות זו *Census* ונסמן אותה ב  $C_{Census}(p, pd)$ .

תיאור פונקציית העלות  $C_{Census}(p, pd)$  – נתבונן בחלון פיקסלים בגודל  $H$  שורות ו  $W$  עמודות מסביב לפיקסל  $I^L(p)$ , הפיקסלים בחלון הגדולים מערך הפיקסל במרכז החלון יקבלו את הערך 1, אחרת יקבלו את הערך 0, תהליך זה נקרא *Census Transform*. התהליך יבוצע באותה הצורה עבור חלון פיקסלים בתמונה הימנית במרכזו של פיקסל  $I^R(pd)$ . שני החלונות יפרשו למערכים חד ממדיים בגודל  $WH$  ותבוצע פעולת XOR בניהם. ערכים זהים יקבלו את הערך 1 וערכים שונים יקבלו את הערך 0. לבסוף תבוצע סְכִימת ערכי המערך לערך אחד, הערך שמתקבל הוא Hamming Distance בין שני החלונות [7].

באיור 8 ניתן לראות דוגמה לשימוש בפונקציית העלות *Census*, ערכים הגדולים מהערך במרכז התמונה יקבלו את הערך 1, אחרת יקבלו את הערך 0.

121	130	26	31	39
109	115	33	40	30
98	102	78	67	45
47	67	32	170	198
39	86	99	159	210

→

1	1	0	0	0
1	1	0	0	0
1	1	X	0	0
0	0	0	1	1
1	1	1	1	1

Census transform window :

1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

איור 7 – דוגמה להפעלת Census Transform על גבי חלון של פיקסלים, שמאל – חלון פיקסלים שבמרכזו פיקסל בעל ערך 78, ימין – חלון המתקבל מהפעלת Census Transform על גבי חלון הפיקסלים השמאלי, כל ערכי הפיקסלים הגדולים מ 78 יקבלו את הערך 1, אחרת יקבלו 0. תחתון – פריסת החלון המתקבל מ Census Transform לכדי מערך חד ממדי.

Left image

1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**XOR**

Right image

1	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Hamming Distance = 3**

0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

איור 8 – דוגמה לשימוש בפונקציית עלות Census המתקבלת על ידי שני חלונות של פיקסלים. עליון – תוצאת Census Transform המתקבלת מחלון פיקסלים בתמונה השמאלית. אמצע - תוצאת Census Transform המתקבלת מחלון פיקסלים בתמונה הימנית. תחתון – פעולת XOR בין תוצאת ה Census Transform מהשורה הראשונה והשנייה, בדוגמה זו  $C_{Census}(p, pd) = 3$ .

יתרון לשימוש בשיטת ה Census הוא עמידות מול שינויים שאינם קשורים למבנה התלת ממדי, אלא קשורים לתמונות. לדוגמה- בהירות שונה לשתי תמונות, רעש בתמונות או תמונות שנלקחו

ממצלמות שונות. יתרון נוסף לשימוש בשיטת ה Census הוא מהירות החישוב ואפשרות לחישוב מקבילי. חסרון השיטה הוא כישלונות באזורים בעלי פרטים החוזרים על עצמם, למשל טקסטורה. באיור 9 ניתן לראות את תמונת העומקים המתקבלת משימוש בפונקציית העלות Census על תמונת הסטראו מאיור 5, בבדיקה מול Ground Truth התקבלה תוצאה של  $PER = 49\%$ .



איור 9 – ימין - מפת העומקים שהתקבלה משימוש בפונקציית עלות Census, אמצע – תמונת העומקים שהתקבלה משימוש בפונקציית העלות AD, שמאל – Ground Truth

### 5.1.3 AD Census

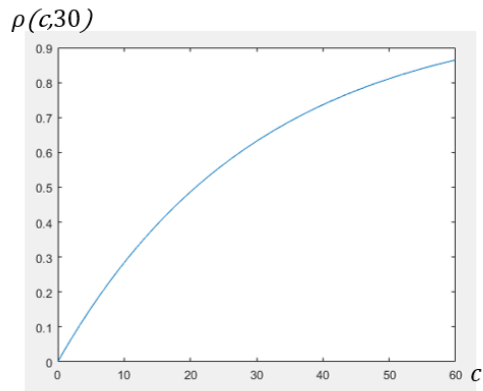
בשני הסעיפים הקודמים הוצגו פונקציות עלות AD ו Census. על מנת להתמודד מול החסרונות שלהן הומצאה שיטה המשלבת בין שתי פונקציות העלות. נקרא לפונקציית העלות המשולבת בשם **AD-Census** ונסמן אותה ב  $C_{AD-Census}(p, pd)$ .

#### משוואה 4

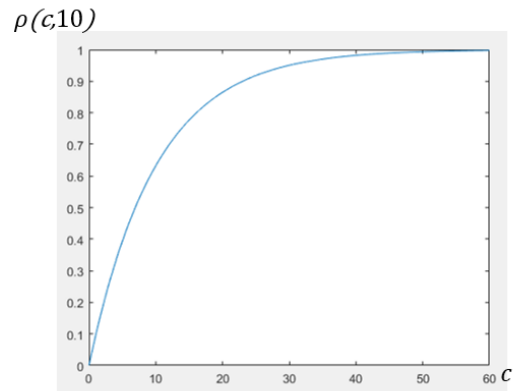
$$C_{AD-Census}(p, pd) = \rho(C_{Census}(p, pd), \lambda_{Census}) + \rho(C_{AD}(p, pd), \lambda_{AD})$$

כאשר  $\rho(c, \lambda)$  המוגדרת במשוואה 4 היא פונקציית נרמול עבור תוצאת פונקציית העלות. פונקציית הנרמול מחזירה ערך בטווח  $[0,1]$ , שואפת אסימפטוטית ל-1, הפרמטר  $\lambda$  שולט בצורת העקמומיות של הפונקציה ו  $c$  היא תוצאת פונקציית העלות הנכנסת לפונקציית הנרמול  $\rho$ . מטרתה של פונקציית הנרמול  $\rho(c, \lambda)$  היא למשקל את פונקציות העלות  $C_{AD}(p, pd)$  ו  $C_{Census}(p, pd)$  לכדי חלק יחסי בתוצאת פונקציית העלות  $C_{AD-Census}(p, pd)$ .

$$\rho(c, \lambda) = 1 - e^{-\frac{c}{\lambda}}$$



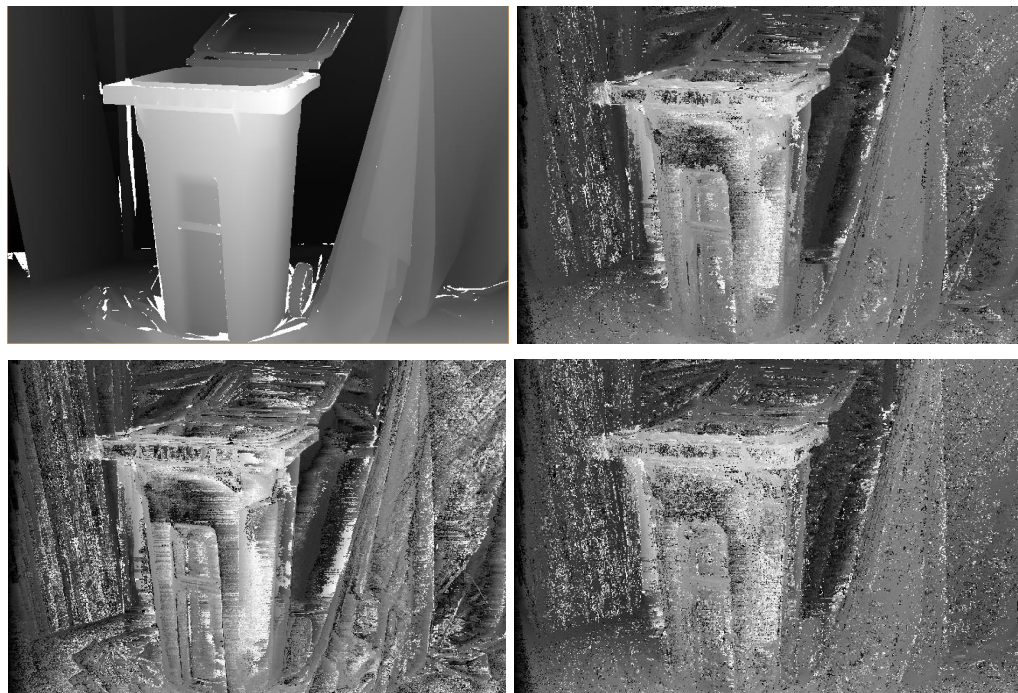
$$\rho(c, \lambda) = 1 - e^{-\frac{c}{30}}$$



$$\rho(c, \lambda) = 1 - e^{-\frac{c}{10}}$$

איור 10 - דוגמה לפונקציה הנרמול  $\rho(c, \lambda)$ , ימין - פונקציית הנרמול  $\rho(c, 10)$  עם ערך ברירת מחדל  $\lambda_{AD} = 10$ , שמאל - פונקציית הנרמול  $\rho(c, 30)$  עם ערך ברירת מחדל  $\lambda_{Census} = 30$

באיור 11 ניתן לראות את תמונת העומקים המתקבלת משימוש בפונקציית העלות AD-Census על תמונות הסטראו מאיור 5. בבדיקה מול Ground Truth התקבלה תוצאה של  $PER = 46\%$ , תוצאה משופרת לעומת כל אחת מפונקציות העלות בנפרד.



איור 11 - ימין עליון - מפת העומקים שהתקבלה משימוש בפונקציית העלות AD-Census, שמאל עליון - Ground Truth, ימין תחתון - מפת העומקים שהתקבלה משימוש בפונקציית עלות Census, שמאל תחתון - תמונת העומקים שהתקבלה משימוש בפונקציית העלות AD



להן פסודו-קוד לחישוב פונקציית העלות AD-Census.

**קלט:** תמונת ימין  $I^R$  ותמונת שמאל  $I^L$ .

**פלט:** פונקציית עלות  $C_{AD-Census}$ .

1. נפעיל את התהליך הבא על כל הפיקסלים בתמונת שמאל, נסמן את הפיקסל בתמונת שמאל ב  $p$ .

1.1.1. נפעיל את התהליך הבא על כל אחד מערכי ה Disparity בטווח החיפוש  $nDisp$ , נסמן את ערך ה Disparity ב  $d$ .

1.1.1.1. נחשב את מיקום הפיקסל התואם לפיקסל  $p$  בתמונה הימינית בהזזה של  $d$  פיקסלים

$$pd = p \pm d$$

1.1.1.2. נחשב את פונקציית העלות  $C_{AD-Census}(p, pd)$

במאמר [5] השתמשו בפרמטרים הבאים: חלון פיקסלים עבור פונקציית העלות Census בגודל  $7 \times 9$ .

$$\lambda_{Census} = 30$$

$$\lambda_{AD} = 10$$

לדעתי השימוש בחלון בגודל  $7 \times 9 = 63$  מהווה איזון אופטימלי בין זמן ריצה לגודל חלון. פעולת מחשב בסיסית, כגון- כפל וחיבור, עובדת על משתנים (בארכיטקטורת מחשב נהוג להשתמש בביטוי אוגר – Register) בגודל של 64 ביט, חלון בגודל 63 הוא אחד מהגדלים המכסימליים לפעולת XOR עבור משתנה בגודל 64 ביט.

פונקציית עלות Census עם חלון בגודל  $7 \times 9 = 63$  מחזירה ערך בטווח  $[0, 63]$ , מטרתו של  $\lambda_{Census} = 30$  היא למשקל בקירוב את פונקציית העלות Census לחצי מפונקציית העלות AD-Census.

## 5.2. שלב ב' Cross Based Cost Aggregation

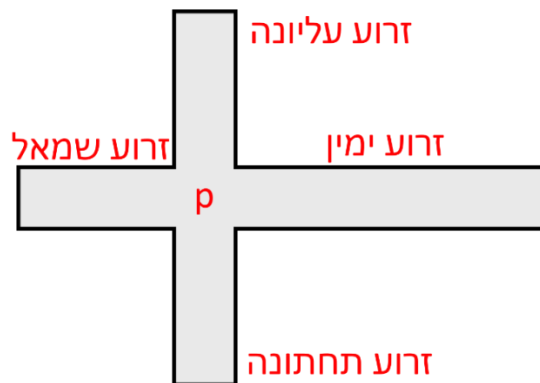
בשלב הקודם הצגנו את פונקציית העלות והראנו כיצד ניתן לחשב את תמונת העומקים בעזרת פונקציית העלות. בסעיף זה תוצג שיטה המשפרת את פונקציית העלות מהשלב הקודם על ידי יוריסטיקת המקום. על פי יוריסטיקת המקום פיקסלים קרובים במרחב בעלי גוון דומה יהיו בעלי אותם ערכי Disparity.

עבור כל פיקסל בתמונה נחשב קבוצת פיקסלים קרובים במרחב ובעלי גוון דומה, קבוצה זו תקרא **אזור הסכימה** (Aggregation Region). אזור הסכימה מחושב בשני שלבים, ראשית תחושב מסגרת אזור הסכימה המהווה קו אופקי וקו אנכי המייצגים את קווי הייסוד של הסגמנט, למסגרת זו נקרא

**שלד ראשוני**. ראה באיור 14 דוגמה לחישוב השלד הראשוני. בשלב הבא יורחב השלד הראשוני לאזור הסכימה באמצעות תהליך שיפורט בהמשך. ראה באיור 15 דוגמה לחישוב אזור הסכימה.

### 5.2.1. חישוב אזור הסכימה

נציג את האלגוריתם לחישוב השלד הראשוני ואזור הסכימה עבור פיקסל יחיד. נציג את האלגוריתם למציאת הקו האופקי מהפיקסל עבורו מחושב אזור הסכימה לכיוון שמאל, הנקרא **הזרוע השמאלית** בשלד הראשוני, ראה איור 12. חישוב השלד הראשוני עבור זרוע ימין, זרוע עליונה וזרוע תחתונה יבוצע בצורה זהה למעט כיווני התקדמות הזרוע.



איור 12 – מבנה השלד הראשוני (קווי המתאר של הסגמנט)

עבור פיקסל  $p$  הזרוע השמאלית מסתיימת כאשר נמצא פיקסל  $p_1$  שאינו מקיים לפחות אחד מהתנאים הבאים:

1. תנאי ראשון

$$D_c(p_1, p) < \tau_1 \text{ and } D_c(p_1, p_1 + (1,0)) < \tau_1$$

כאשר  $D_c(p_1, p)$  המוגדר במשוואה 5 הוא הפרש הגוונים בין פיקסל  $p$  לפיקסל  $p_1$ . מכיוון שמתבצעת הרחבת הזרוע לכיוון שמאל,  $p_1 + (1,0)$  הוא הפיקסל הקודם לפיקסל  $p_1$  על כיוון התקדמות הזרוע.

### משוואה 5

$$D_c(p_1, p) = \max_{i=R,G,B} |I_i(p_1) - I_i(p)|$$

התנאי הראשון מגביל את הפרש הגוונים בין פיקסל  $p_1$  לפיקסל  $p$ . החלק השני של התנאי  $D_c(p_1, p_1 + (1,0)) < \tau_1$  מוודא כי הגוון באזור הסכימה נשאר דומה.



2. תנאי שני

$$D_s(p_1, p) < L_1$$

$D_s(p_1, p)$  מוגדר כמרחק המרחבי בין פיקסל  $p_1$  לפיקסל  $p$ .

$$D_s(p_1, p) = |p_1 - p|$$

התנאי השני מגביל את התרחבות הזרוע למרחק מכסימלי של  $L_1$  פיקסלים בין פיקסל  $p_1$  לפיקסל  $p$ .

3. תנאי שלישי

$$\text{if } L_2 < D_s(p_1, p) < L_1 \rightarrow D_c(p_1, p) < \tau_2$$

התנאי השלישי מגביל את הפרשי הגוונים בין פיקסל  $p_1$  לבין פיקסל  $p$  כאשר המרחק בין פיקסל  $p_1$  לבין פיקסל  $p$  גדול מערך הסף  $L_2$ .

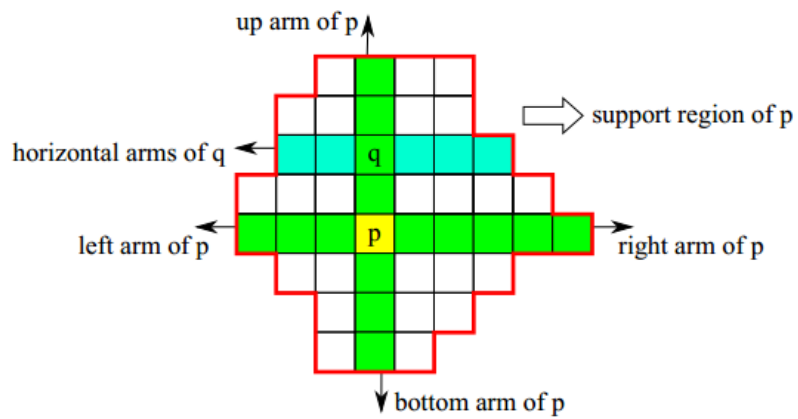
הרעיון המרכזי העומד מאחורי תנאי זה הוא שליטה גמישה בהתרחבות אזור הסכימה, בשלב בו אזור הסכימה מתרחב מעבר למרחק הנקבע על ידי ערך הסף  $L_2$  ( $L_2 < L_1$ ), משתמשים בסף הדוק יותר  $\tau_2$  ( $\tau_2 < \tau_1$ ). תנאי זה מאפשר את התרחבות הזרוע במידה והגוונים מאוד דומים.

נסמן את  $\tau_2, \tau_1$  כשני ספים המייצגים את הפרשי הגוונים המכסימלי בין פיקסל  $p$  לפיקסל  $p_1$ . נסמן את  $L_2, L_1$  כשני ספים המייצגים את המרחק המכסימלי בין פיקסל  $p$  לפיקסל  $p_1$ .

השלד הראשוני נבנה על ידי תהליך דומה בכיוונים שונים על הזרוע הימנית, הזרוע התחתונה והזרוע העליונה.

השלד הראשוני יורחב לאזור הסכימה על ידי מעבר כל השורות בשלד הראשוני והרחבת כל שורה ימינה ושמאלה באמצעות שלושת תנאים אלו.

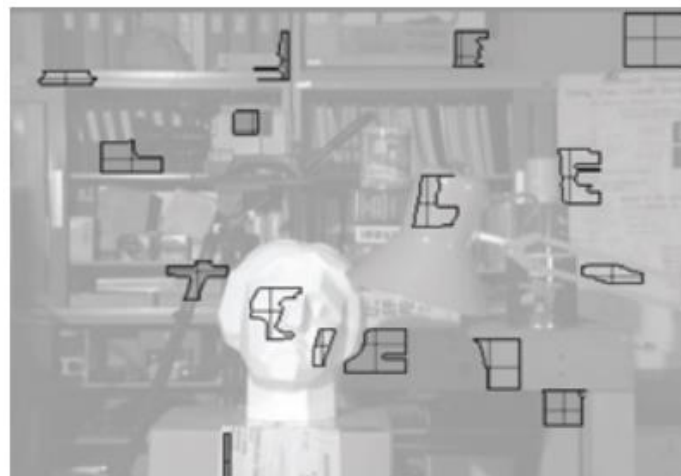
באיור 13 ניתן לראות את אזור הסכימה במסגרת אדומה, השלד הראשוני בירוק ודוגמה להרחבת השלד הראשוני לאזור הסכימה בתכלת.



איור 13 – דוגמה לחישוב אזור הסכימה מתוך השלד הראשוני בעזרת סביבה של פיקסלים



איור 14 – מספר דוגמאות לשלד הראשוני (קווי המתאר של הסגמנט) המתקבל עבור סביבה של פיקסלים. כל מקטע פיקסלים בצבע שחור בצורת פלוס הוא דוגמה לשלד ראשוני



איור 15 – מספר דוגמאות לאזור הסכימה המתקבל עבור סביבה של פיקסלים. כל צורה גיאומטרית בצבע שחור הינה דוגמה לאזור הסכימה עבור הפיקסל שנמצא במרכז מקטע הפיקסלים

בשלב זה נמצע את פונקציית העלות באזור הסכימה שחושב בשלב הקודם. מיצוע פונקציית העלות מבוצע על ידי סכימת ערכי הפונקציה באזור הסכימה וחלוקה במספר האיברים. על מנת לבצע את הסכימה ביעילות נשתמש בשיטה הנקראת *Integral Image* [8].

## 5.2.2.1 Integral Image

Integral Image הוא אלגוריתם לחישוב מהיר ויעיל של סכומי תתי קבוצות מלבניות ברשת קואורדינטות דיסקרטיות – לדוגמה תמונה.

**האלגוריתם** – הערך בכל נקודה  $(x, y)$  בטבלת ה Integral Image הוא סכום כל הפיקסלים מעל ומשמאל לנקודה  $(x, y)$  כולל

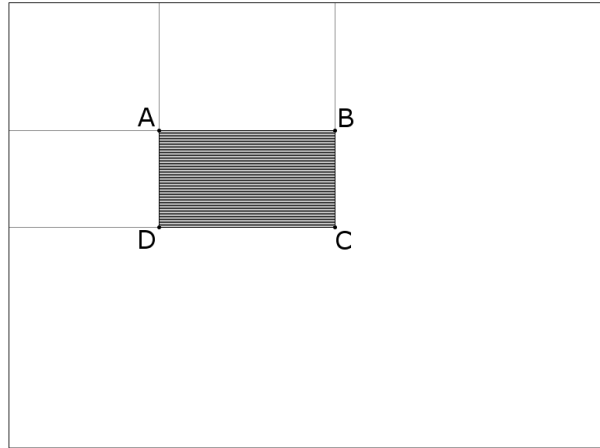
$$sum(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} I(x', y')$$

ניתן לחשב את טבלת ה Integral Image באופן יעיל על ידי מעבר יחיד על התמונה, תוך שימוש בעובדה שהערך בטבלה בנקודה  $(x, y)$  הוא למעשה:

$$sum(x, y) = I(x, y) + sum(x - 1, y) + sum(x, y - 1) - sum(x - 1, y - 1)$$

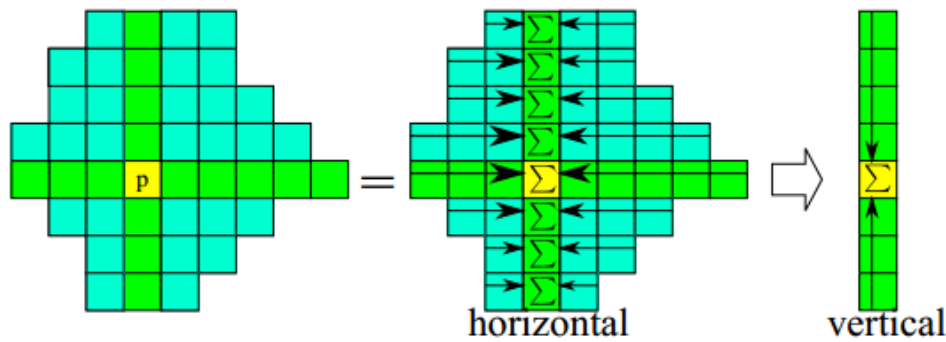
לאחר שטבלת ה Integral Image חושבה, משימת חישוב סכומי מלבנים בתמונה יכולה להתבצע בזמן קבוע, תוך דגימת ארבעה ערכים בלבד מהטבלה. לדוגמה סכום הערכים במלבן המוצג באיור 16 הוא:

$$\sum_{\substack{A(x) < x' \leq C(x) \\ A(x) < y' \leq C(y)}} I(x', y') = sum(A) + sum(C) - sum(B) - sum(D)$$



איור 16 – חישוב שטח בתחום מלבני באמצעות Integral Image

נשוב למיצוע אזור הסכימה. כל שורה באזור הסכימה תסכם בנפרד באמצעות Integral Image. לאחר מכן נסכום של כל סכומי השורות לכדי ערך אחד, ראה דוגמה לשיטת הסכימה באיור 17.



איור 17 - דוגמה למתודולוגיה מיצוע אזור הסכימה. שמאל – אזור הסכימה, אמצע – סכימת כל שורה בנפרד, ימין – סכימת כל סכומי השורות לערך אחד

להן פסודו-קוד לחישוב שלב ב' Cross Based Cost Aggregation.

**קלט:** פונקציית עלות  $C_{AD-Census}$  שחושבה בשלב הקודם.

**פלט:** פונקציית עלות משופרת  $C_{Cost Aggregation}$ .

1. נחשב את אזור הסכימה עבור כל אחד מהפיקסלים בתמונת שמאל.

2. נשפר את פונקציית העלות  $C_{AD-Census}$  על ידי מיצוע אזור הסכימה:

נפעיל את התהליך הבא על כל הפיקסלים בתמונת שמאל, נסמן את הפיקסל בתמונת שמאל ב

$p$ .

2.1. נפעיל את התהליך הבא על כל אחד מערכי ה Disparity בטווח החיפוש  $nDisp$ , נסמן את

ערך ה Disparity ב  $d$ .

2.1.1. נחשב את מיקום הפיקסל התואם ל  $p$  בתמונה הימינית בהזזה של  $d$  פיקסלים

$$pd = p \pm d$$

2.1.2. נמצע את פונקציית העלות  $C_{AD-Census}(p, pd)$  באזור הסכימה שחושב עבור פיקסל

$p$  ונציב את התוצאה ב  $C_{Cost Aggregation}(p, pd)$

מחברי המאמר [5] מציעים להשתמש בפרמטרים הבאים:

$$L_1 = 34$$

$$L_2 = 17$$

$$\tau_1 = 20$$

$$\tau_2 = 6$$

לדעתי ספים הללו נקבעו באמצעות ניסוי וטעיה ועתידיים להשתנות בין תמונות סטראו שונות. באיור 18 ניתן לראות את תמונת העומקים המתקבלת משלב ב' של האלגוריתם עבור תמונות הסטראו מאיור 5. בבדיקה מול Ground Truth התקבלה תוצאה של  $PER = 20\%$ , תוצאה המעידה על שיפור משמעותי לעומת התוצאה המתקבלת באיור 11 מהשלב הקודם.



איור 18 – ימין - מפת העומקים שהתקבלה משימוש בפונקציית העלות משלב ב' Cross Based Cost Aggregation, אמצע - מפת העומקים שהתקבלה משימוש בפונקציית העלות משלב א' AD-Census, שמאל - Ground Truth

### 5.3. שלב ג' Semi Global Matching

#### 5.3.1. מוטיבציה

חישוב ערכי ה Disparity מפונקציית העלות רחוקה מלהיות נטולת טעויות, ערכי ה Disparity יכולים להיות שגויים עקב שימוש בפונקציית עלות לא מתאימה, הבדלים קלים במצלמות, רעש, פרמטרים לא נכונים ומקרים שבהם היוריסטיקה לא הייתה מספיק מדויקת.

בשלב זה נשפר את פונקציית העלות שהתקבלה בשלב הקודם כך שערכי ה Disparity שנקבל בתמונת העומקים לא ינועו מעלה ומטה בצורה שאינה מייצגת את המרחב התלת ממדי.

נרצה לעדכן את פונקציית העלות באמצעות תנאי שיקשה על שינויי ערכי ה Disparity בין פיקסלים שכנים בפונקציית העלות המתוקנת, על ידי פעולה זו נגרום להחלקת ערכי ה Disparity בתמונת העומקים המתקבלת כפלט עבור אלגוריתם להתאמת סטראו. פונקציית העלות ותנאי החלקת רציפות ה Disparity מבוטאים באמצעות משוואת אנרגיה  $E(D)$  התלויה בפונקציית העלות. נסמן את תמונת העומקים ב  $D$ .

משוואה 6

$$\operatorname{argmin}_D(E(D)) = \sum_p \left( C(p, D(p)) + \sum_{q \in N_p} P_1 T_1(|D(p) - D(q)|) + \sum_{q \in N_p} P_2 T_2(|D(p) - D(q)|) \right)$$

כאשר הפונקציות  $T_1$  ו  $T_2$  מוגדרות להלן

$$T_1(x) = \begin{cases} 1, & x = 1 \\ 0, & \text{Otherwise} \end{cases}$$

$$T_2(x) = \begin{cases} 1, & x > 1 \\ 0, & \text{Otherwise} \end{cases}$$

נסמן ב  $D(p)$  את ערך ה Disparity  $d$  עבור פיקסל  $p$  בתמונת העומקים  $D$ , ונסמן את סביבת הפיקסלים השכנים לפיקסל  $p$  על ידי  $N_p$ . הביטוי הראשון במשוואה 6 מבצע סכימה של פונקציית העלות עבור פיקסל  $p$  ו Disparity  $D(p)$ . הביטוי השני מוסיף קנס קטן  $P_1$  למשוואת האנרגיה עבור כל פיקסל השכן לפיקסל  $p$  כך שהפרש ערך ה Disparity בניהם שונה בפיקסל אחד לכל היותר. הביטוי השלישי מוסיף קנס גדול יותר  $P_2$  למשוואת האנרגיה עבור כל פיקסל השכן לפיקסל  $p$  כך שהפרש ערך ה Disparity בניהם גדול מעבר לפיקסל אחד ( $P_2 \geq P_1$ ).

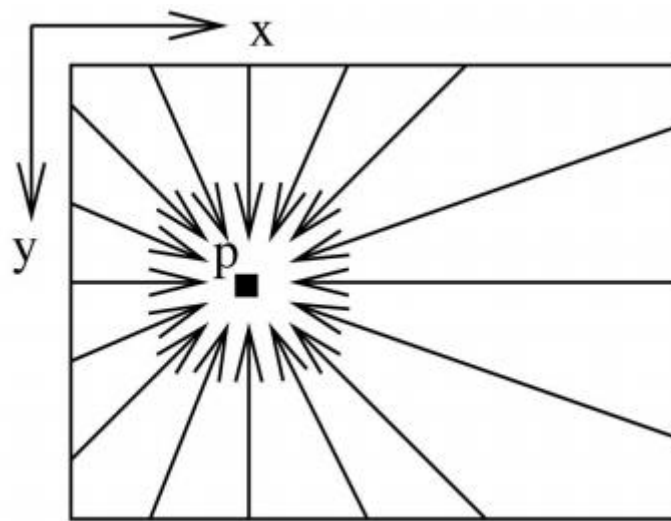
כעת ניתן לפתור את התאמת הסטראו על ידי מציאת תמונת עומקים  $D$  הממזערת את משוואת האנרגיה  $\operatorname{argmin}_D(E(D))$ . אולם אופטימיזציה גלובלית שכזו עבור  $\operatorname{argmin}_D(E(D))$  בשני ממדים הינה בעיית  $NP$  Complete [9] ואינה ניתנת לפתרון בזמן סביר. לעומת זאת מינימיזציה של משוואת אנרגיה בחד ממד ניתנת לפתרון בזמן פולינומיאלי באמצעות Dynamic Programming [10].

### 5.3.2 אלגוריתם Semi Global Matching

הרעיון באלגוריתם Semi Global Matching הוא שילוב בין מספר משוואות אופטימיזציה חד ממדיות על מנת לקבל פתרון מקורב לפתרון אופטימיזציה גלובלי.

אלגוריתם Semi Global Matching מפעיל מספר משוואות אופטימיזציה חד ממדיות בכיוונים שונים, כל משוואת אופטימיזציה סוכמת את ערכי פונקציית העלות על פי חוקיות שתפורט בהמשך. התוצאה של הפעלת משוואות אופטימיזציה על פונקציית העלות היא **פונקציית עלות מוחלקת**, נקרא להפעלת משוואות האופטימיזציה בכיוון יחיד **אופטימיזציה קו סריקה** (Scanline Optimization).

הפלט של אלגוריתם **Semi Global Matching** הוא פונקציית עלות מוחלקת ומשופרת  $C_{SGM}(p, d)$ .  $C_{SGM}(p, d)$  מחושבת עבור פיקסל  $p$  ו Disparity  $d$  על ידי מיצוע פונקציות העלויות המוחלקות מכיוונים שונים המסתיימות בפיקסל  $p$  ו Disparity  $d$ , ראה דוגמה באיור 19.



איור 19 – פונקציית העלות SGM עבור פיקסל  $p$  המתקבלת משימוש ב 16 קווי סריקה מכיוונים שונים [11]

נגדיר מספר משוואות אופטימיזציה בכיוונים שונים, נסמן את כיוון משוואת האופטימיזציה ב  $r$  ונקרא לפיקסל הקודם לפיקסל  $p$  על כיוון משוואת האופטימיזציה ב  $p_r^{-1}$ . בשונה מהשלבים הקודמים נסמן את **פונקציית העלות** עבור פיקסל  $p$  ועבור Disparity  $d$  ב  $C_2(p, d)$  (סימון זה הוא תוספת לייצוג הקודם  $C(p, pd)$ , ניתן להמיר בין הייצוגים על ידי  $(C(p, pd) = C_2(p, \pm(pd - p)))$ . נסמן את פונקציית העלות המתקבלת על ידי משוואת האופטימיזציה בכיוון  $r$  על ידי  $C_r(p, d)$  ונגדיר את משוואת האופטימיזציה במשוואה 7.

משוואה 7 - משוואת אופטימיזציה קו סריקה

$$C_r(p, d) = C_2(p, d) + \left[ \min \left\{ \begin{array}{l} C_r(p_r^{-1}, d) \\ C_r(p_r^{-1}, d \pm 1) + P_1 \\ \min_k C_r(p_r^{-1}, k) + P_2 \end{array} \right\} - \min_k C_r(p_r^{-1}, k) \right]$$

כאשר ערכי הסף  $P_1$  ו  $P_2$  המוגדרים במשוואה 7 משמשים כקנסות (Penalty) במשוואת האופטימיזציה ונקבעים על ידי הפרשי הגוונים בתמונה הימנית ובתמונה השמאלית על פי התנאים הבאים

$$D_1 = D_c(p, p_r^{-1})$$

$$D_2 = D_c(pd, pd_r^{-1})$$

$$D_c(p_1, p_2) = \max_{i=R,G,B} |I_i(p_1) - I_i(p_2)|$$

א. אם  $D_1 < \tau_{SO}$  וגם  $D_2 < \tau_{SO}$  אזי  $P_1 = \chi_1, P_2 = \chi_2$

ב. אם  $D_1 < \tau_{SO}$  וגם  $D_2 > \tau_{SO}$  אזי  $P_1 = \frac{\chi_1}{4}, P_2 = \frac{\chi_2}{4}$

ג. אם  $D_1 > \tau_{SO}$  וגם  $D_2 < \tau_{SO}$  אזי  $P_1 = \frac{\chi_1}{4}, P_2 = \frac{\chi_2}{4}$

ד. אחרת  $P_1 = \frac{\chi_1}{10}, P_2 = \frac{\chi_2}{10}$

נסמן את  $\chi_1$  ו  $\chi_2$  כקבועים ונסמן את  $\tau_{SO}$  כסף הפרשי הגוונים. פונקציית העלות  $C_{SGM}(p, d)$  תתקבל על ידי מיצוע פונקציות העלויות מכיווני הסריקה השונים.

$$C_{SGM}(p, d) = \frac{1}{R} \sum_{r=1}^R C_r(p, d)$$

להלן פסודו-קוד לחישוב שיטת אופטימיזצית קו סריקה.

**קלט:** פונקציית עלות  $C_{Cost Aggregation}$  שחושבה בשלב הקודם.

**פלט:** פונקציית עלות משופרת  $C_{SGM}$ .

1. נפעיל את התהליך הבא על כל אחד מכיווני הסריקה, נסמן את כיוון הסריקה ב  $r$ .

1.1. נפעיל את התהליך הבא על כל הפיקסלים בתמונת שמאל, נסמן את הפיקסל בתמונת

שמאל ב  $p$ .

1.1.1. נחשב את פונקציית העלות המוחלקת המתקבלת על ידי אופטימיזצית קו סריקה

בכיוון  $r$  באמצעות משוואה 7 ונשמור את התוצאה ב  $C_r(p, d)$ .

2. נפעיל את התהליך הבא על כל הפיקסלים בתמונת שמאל, נסמן את הפיקסל בתמונת שמאל ב

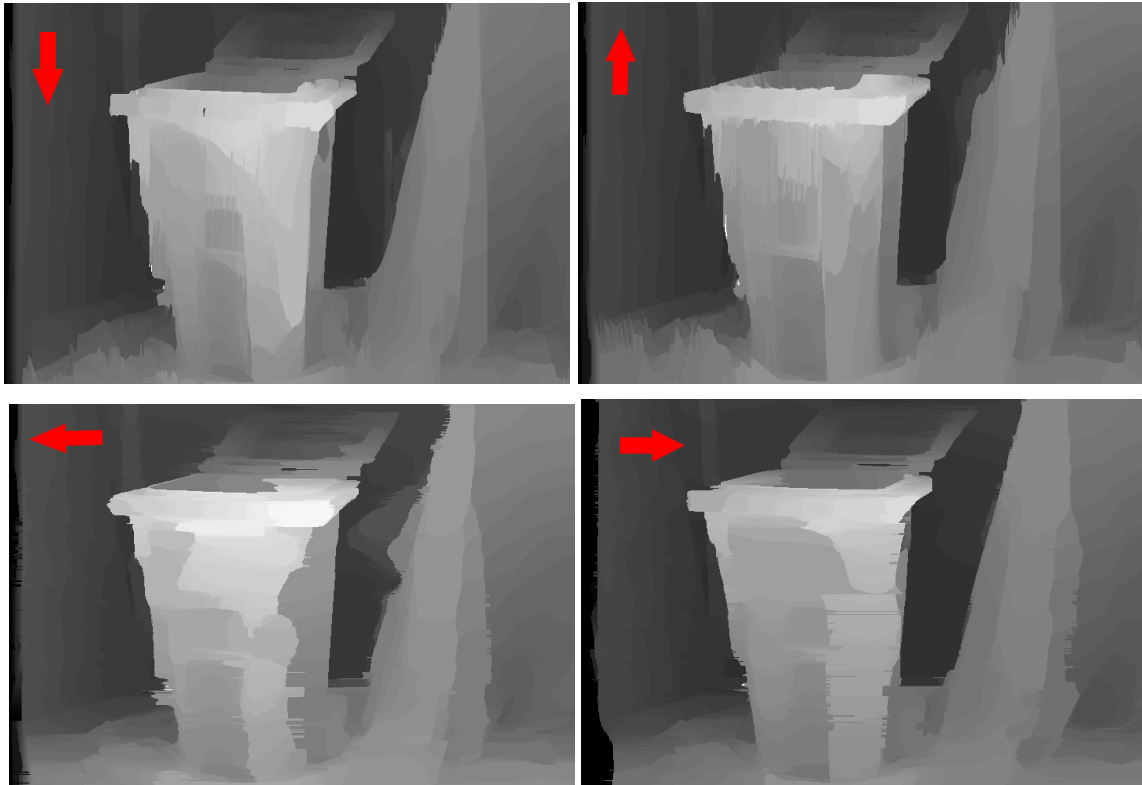
$p$ .

2.1. נמצע את פונקציות העלויות המוחלקות

$$C_{SGM}(p, d) = \frac{1}{R} \sum_{r=1}^R C_r(p, d)$$



באיור 20 ניתן לראות את תמונות העומקים המתקבלות לאחר אופטימיזצית קו סריקה בכיוונים שונים. בכל התמונות ניתן לראות קווי רציפות בכיוון הסריקה כתוצאה מפעולת ההחלקה וקווי אי רציפות שלא בכיוון הסריקה.



איור 20 - תמונות העומקים שהתקבלו משימוש באלגוריתם קו סריקה בכיוונים שונים. ימין עליון – תמונת העומקים המתקבלת לאחר אופטימיזצית קו סריקה אנכי מלמטה למעלה, שמאל עליון – תמונת העומקים המתקבלת לאחר אופטימיזצית קו סריקה אנכי מלמעלה למטה, ימין תחתון – תמונת העומקים המתקבלת לאחר אופטימיזצית קו סריקה אופקי משמאל לימין, ושמאל תחתון – תמונת העומקים המתקבלת לאחר אופטימיזצית קו סריקה אופקי מימין לשמאל

שיטת אופטימיזצית קו סריקה משפרת את תוצאות התאמת הסטראו כתוצאה מהכנסת תנאי רציפות ה Disparity בכיוונים שונים. חסרון השיטה הוא זמן ריצה גבוה עקב חישוב כיווני סריקה שונים, חלק מכיווני הסריקה אינם עולים בקנה אחד עם ארכיטקטורת זיכרון של מחשב וגורמים לאיטיות. חיסרון נוסף הוא סיבוכיות זיכרון גבוהה, מכיוון שיש לשמור את תוצאות כל כיווני הסריקה עד לשלב המיצוע.

כמות כיווני קווי הסריקה כפי שהוצגו במאמר המקורי [11] הייתה 16 כיווני סריקה ולא פחות מ 8 כיווני סריקה. בשיטת AD - Census [5] הוחלט להשתמש ב 4 כיווני סריקה על מנת לקצר את זמן הריצה, שני כיווני סריקה אופקיים ושני כיווני סריקה אנכיים  $\updownarrow \rightarrow \leftarrow$ .

באיור 21 ניתן לראות את תמונת העומקים המתקבלת כתוצאה מהפעלת משוואה 3 על פונקציית העלות המתקבלת מאלגוריתם Semi Global Matching.



איור 21 – ימין - תמונת העומקים שהתקבלה משימוש באלגוריתם Semi Global Matching, אמצע - מפת העומקים שהתקבלה משימוש ב Cross based cost aggregation, שמאל - Ground Truth

בסיום שלב זה, נחשב את תמונת העומקים המתקבלת מפונקציית העלות  $C_{SGM}(p, d)$  באמצעות משוואה 3. משלב זה ואילך, פעולות חישוב מבוצעות על תמונת העומקים.

#### 5.4. שלב ד' Multistep Disparity Refinement

עד עכשיו תוארו אלגוריתמים לתיקון פונקציית העלות המתבססים על יוריסטיקת הרציפות, אלגוריתם Multistep Disparity Refinement מיועד לתקן אזורים בהם פונקציית העלות אינה רציפה בהגדרה, אזורים בעלי שינוי גוון חד (edge) בתמונת הסטראו או אזורים מוסתרים בתמונת הסטראו.

מטרת שלב זה היא שיפור תמונת העומקים שהתקבלה בשלב הקודם. האלגוריתם מורכב משני שלבים, בשלב הראשון גילוי פיקסלים בעלי ערכי Disparity שגויים, ובשלב השני סדרת אלגוריתמים לתיקון המופעלים סדרתית.

##### 5.4.1 Outlier Detection

מטרת אלגוריתם Outlier Detection הוא מציאת קבוצת פיקסלים שגויים, קבוצת זו תשמש כקבוצת פיקסלים המועמדת לתיקון ערכי ה Disparity.

נפעיל אלגוריתם להתאמת סטראו משלב א' עד שלב ג', כל פיקסל בתמונת ימין יותאם לפיקסל בתמונת שמאל, נסמן את תמונת העומקים המתקבלת ב  $D_R$ . נפעיל את האלגוריתם שוב כך שכל פיקסל בתמונת שמאל יותאם לפיקסל בתמונת ימין, נסמן את תמונת העומקים המתקבלת ב  $D_L$ . כעת נוודא את נכונות התאמת ערכי ה Disparity משני הכיוונים.

נסמן את ערך ה Disparity של פיקסל  $p$  כשגוי אם

$$D_L(p) \neq D_R(p - D_L(p))$$

מטרת שלב זה היא תיקון ערכי ה Disparity מקבוצת הפיקסלים השגויים בעזרת ערכי ה Disparity במרחב. עבור כל פיקסל מקבוצת ה Disparity השגויים נבנה היסטוגרמה  $H_p$  של ערכי ה Disparity התקינים באזור הסכימה שחושב בשלב ב'. נסמן את כמות התאים המכסימלית בהיסטוגרמה ב  $d_{max} + 1$ , נסמן את התא המקסימלי בהיסטוגרמה ע"י  $d_p^*$  וגודל ההיסטוגרמה ב

$$S_p = \sum_{d=0}^{d_{max}} H_p(d)$$

ערך ה Disparity של פיקסל שגוי  $p$  יעודכן לערך Disparity מתוקן  $d_p^*$  אם מתקיימים התנאים הבאים:

$$S_p > \tau_s$$

$$\frac{H_p(d_p^*)}{S_p} > \tau_H$$

על מנת לתקן כמה שיותר ערכי Disparity התהליך מבוצע 5 פעמים, כאשר ערכי ה Disparity המתוקנים משמשים כערכים תקינים עבור השלב הבא. מחברי המאמר [5] מציעים את הפרמטרים הבאים

$$\tau_s = 20$$

$$\tau_H = 0.4$$

ערכי ה Disparity שלא תוקנו בשלב הנוכחי ממשיכים לשלב הבא.

מטרת אלגוריתם Proper Interpolation היא תיקון ערכי ה Disparity מקבוצת הפיקסלים השגויים באמצעות אינטרפולציה של ערכי Disparity תקינים וקרובים במרחב. עבור כל פיקסל בעל ערך Disparity מקבוצת הפיקסלים השגויים נבצע אינטרפולציה ב 16 כיוונים שונים, נבחר את ערך ה Disparity המתוקן כערך ה Disparity של הפיקסל בעל הצבע הקרוב ביותר לפיקסל עבורו אנו מבצעים את התיקון.

התהליך מבוצע מספר פעמים כאשר ערכי ה Disparity המתוקנים משמשים כערכים תקינים עבור השלב הבא. ערכי העומקים שלא תוקנו בשלב הנוכחי חוזרים לערכם המקורי.

	10		11	
9	1	2	3	12
	8	P	4	
16	7	6	5	13
	15		14	

איור 22 – כיווני האינטרפולציה באלגוריתם Proper Interpolation. באיור מוצגים 16 כיווני האינטרפולציה המסומנים בספרות מ 1 עד 16 המשמשים לתיקון ערך ה Disparity של פיקסל  $p$  המופיע במרכז התמונה.

#### 5.4.4 Sub Pixel Enhancement

אלגוריתם Sub Pixel Enhancement משפר את דיוק תמונת העומקים המיוצגת במספרים שלמים לתמונת עומקים המיוצגת במספרים רציונליים באמצעות שיערוך תת פיקסלי. האלגוריתם משמש להורדת שגיאות הנובעות מערכי Disparity בדידים ומבוסס על Quadric Polynomial Interpolation. נעבור על כל הפיקסלים בתמונת העומקים, ערך ה Disparity של כל פיקסל ישופר באמצעות משוואה 8. נסמן את פונקציית העלות משלב ג' ב  $C_2(p, d)$  ונסמן את ערך ה Disparity המשופר על ידי  $d^*$ .

משוואה 8

$$d^* = d - \frac{C_2(p, d_+) - C_2(p, d_-)}{2(C_2(p, d_+) + C_2(p, d_-) - 2C_2(p, d))}$$

$$d = D_L(p)$$

$$d_+ = d + 1$$

$$d_- = d - 1$$

במטרה להסיר רעשים קטנים תמונת העומקים הסופית מתקבלת על ידי החלקת תמונת העומקים שהתקבלה על ידי אלגוריתם Sub Pixel Enhancement על ידי **מסנן חציון** (Median Filter) בגודל  $3 \times 3$ . שלב ד' משפר את תוצאות האלגוריתם ב  $PER = 3.8\%$  [5].

באיור 23 ניתן לראות את תמונת העומקים שהתקבלה בשלב ד' Multistep Disparity Refinement של האלגוריתם על תמונות הסטראו מאיור 5. בבדיקה מול Ground Truth התקבלה תוצאה של  $PER = 25\%$ .



איור 23 – ימין - תמונת העומקים שהתקבלה משימוש בשלב ד' של האלגוריתם, אמצע - תמונת העומקים המתקבלת לאחר הפעלת אלגוריתם קו סריקה, שמאל - Ground Truth

## 6. מימוש אלגוריתם AD-Census

### 6.1 מימוש האלגוריתם ב C++ ו Matlab

האלגוריתם נכתב ב C++ משיקולי זמן ריצה. C++ היא שפת תכנות בעלת פקודות מכונה מהירות ויכולות שליטה על מרחבי זיכרון והיא השפה המקובלת ביותר לכתיבת תוכנות אלגוריתמיות. בכדי לאפשר למשתמש אינטראקציה מול האלגוריתם תוכנן ומומש **ממשק משתמש**, Graphical User Interface (GUI) של Matlab. קוד ה C++ קומפל לספריה בינרית (DLL) ובכך התאפשרה הפעלת האלגוריתם מתוך סביבת ה Matlab.

#### 6.1.1 סביבה

- הפרויקט מומש בסביבת Windows 10.
- קוד ה C++ קומפל באמצעות Microsoft Visual Studio 2015.
- Matlab 2017A

#### 6.1.2 ספריות הייצונית

כמקובל בעולם התוכנה הפרויקט מומש בעזרת ספריות מבוססות קוד פתוח. בפרט,

1. OpenCV – Open Source Computer Vision

ספריית קוד פתוח המכילה אלגוריתמים בעיבוד תמונה וראיה ממוחשבת.

2. Concurrency Library – The Parallel Patterns Library

ספריה התומכת בעיבוד מקבילי (יוגדר בהמשך).

תוכנה המאפשרת להציג את תוצאת התאמת הסטראו כמודל תלת ממדי.

### 6.1.3. מקבילות המערכת

**עיבוד מקבילי** הוא מונח המציין עיבוד בו־זמני של מטלה מסוימת על ידי מספר מעבדים. המטלה מחולקת בין המעבדים כדי להגיע לתוצאות מהר יותר מאשר בעיבוד טורי. עיבוד מקבילי מבוסס על פיצול תהליך הפתרון של בעיה כלשהי למספר מטלות קטנות יותר, שאותן ניתן לבצע בו־זמנית, עם מידה מסוימת של **תיאום** (סנכרון[synchronize] [12]).

במהלך המימוש התגלו מספר אלגוריתמים הניתנים לעיבוד מקבילי, לכן הוחלט לנצל מעבד מרובה ליבות, Multi Core Processor, בכדי להאיץ את זמני הריצה של האלגוריתם.

נסווג את העיבוד המקבילי לשני סוגים. הראשון מיקבול שלבים אלגוריתמיים שונים בתהליך, למשל שלב ה- Outlier Detection של האלגוריתם מחייב הרצה של שלבי האלגוריתם א', ב' ו-ג', מתמונת ימין לתמונת שמאל ותמונת שמאל לתמונת ימין, פעולות אלו מורצות במקביל.

הסוג השני הוא עיבוד מקבילי של פעולה אלגוריתמית זהה על ערכים שונים, Single Instruction Multiple Data (SIMD). לדוגמה חישוב AD Census, בפעולה טורית יבוצע מעבר על כל הפיקסלים בתמונה - מעבר על כל שורה ועל כל עמודה כל אחת בתורה. לעומת זאת בפעולה מקבילית כל יחידה מקבילית (Thread) תבצע את חישוב ה AD Census עבור שורה אחת או יותר במקביל ליחידות האחרות.

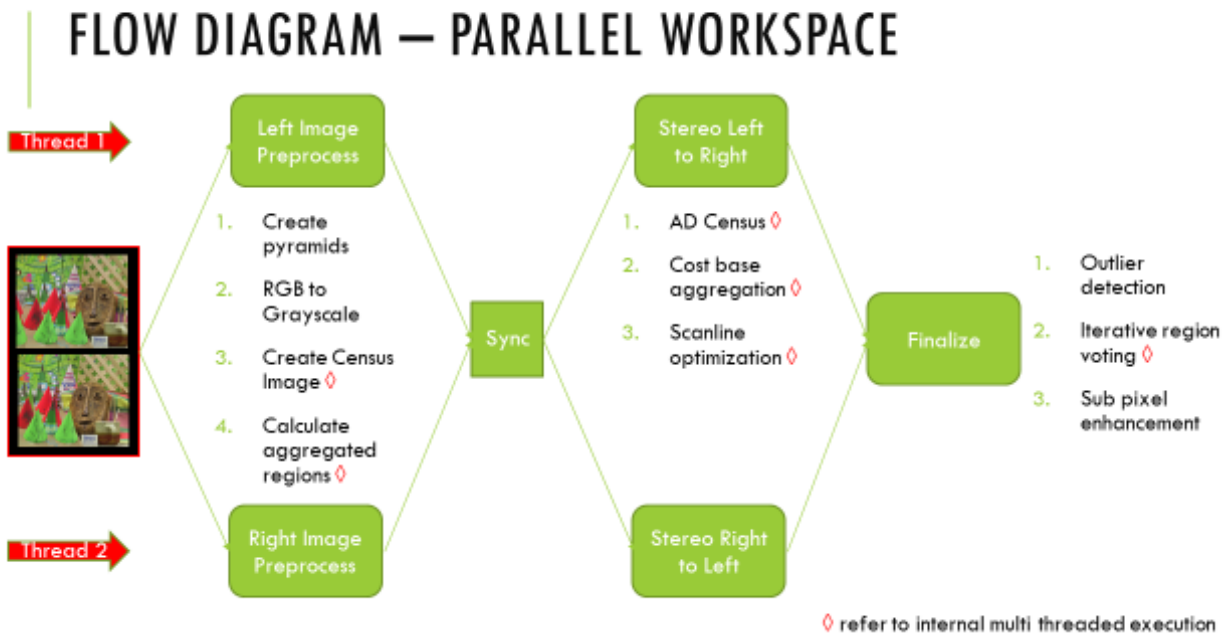
נציין כי עיבוד מקבילי היא פעולה עדינה ומסובכת. עודף פעולות מקבול עשוי לגרום לאיטיות המערכת, אולם חוסר מקבול מונע ניצול מיטבי של החומרה ופגיעה בזמן ריצה.

באיור 24 ניתן לראות את תרשים העיבוד המקבילי של האלגוריתם. נסביר את התרשים משמאל לימין - האלגוריתם מקבל שתי תמונות ובאמצעות שני **תהליכונים** (threads) מחשב עבור כל תמונה:

יצירת פירמידות, המרת התמונה מגווני צבע לגווני אפור, ייצוג כל פיקסל בתמונה ב Census Transform וחישוב אזור הסכימה. לאחר מכן, מתבצע סנכרון בין שני התהליכונים המסומן ע"י ריבוע בשם Sync. בהמשך מופעל האלגוריתם להתאמת סטראו במקביל מתמונת שמאל לתמונת ימין ומתמונת ימין לתמונת שמאל על ידי שני תהליכונים. כל אחד מהתהליכונים מחשב את שלב א', ב' וג' של האלגוריתם. בהמשך מבוצע סנכרון ולאחר מכן מחושבות תתי הפעולות של שלב ד' Multistep

Disparity Refinement

התפצלות לשני חיצים מסמלת התפצלות לשני תהליכונים שונים, כלומר עיבוד מקבילי מסוג ראשון. קובייה אלגוריתמית המסומנת על ידי מעוין אדום ניתנת לעיבוד מקבילי פנימי, כלומר מיקבול מסוג שני.



איור 24 – תרשים עיבוד מקבילי של האלגוריתם

## 6.2. סיבוכיות זיכרון

אלגוריתם להתאמת סטראו דורש שימוש במשטחי זיכרון על מנת לשמור את תוצרי שלבי האלגוריתם. משיקולי דיוק נייצג כל פיקסל בתמונה על ידי משתנה מסוג Float, בייצוג זה גודלו של כל פיקסל הוא 4 byte, נסמן את היחס בין גודל הזיכרון לפיקסל על ידי  $P_{size} = 4 \frac{byte}{pixel}$ , נסמן את  $C$  ככמות סופית של משטחי זיכרון עזר (באופן גס  $C = 7$ ). הזיכרון הדרוש להרצת האלגוריתם מסומן ב  $Memory$  ומיוצג ביחידות של  $P_{size}$ .

$$Memory = 2(MN(nDisp + C)) \quad [P_{size}]$$

[מכפלת המשוואה ב 2 מופיעה עקב שלב Outlier Detection של האלגוריתם].

לדוגמה- מציאת התאמת סטראו עבור תמונת Full HD בגודל  $1920 \times 1080$ , כאשר  $nDisp = 160$ , דורשת הקצאת זיכרון של לפחות

$$2 * (1920 * 1080 * (160 + 7) * 4) = 2461MB$$

### 6.3 מבנה מודולי תוכנה

בשלב זה יפורט מבנה מודולי התוכנה (UML) Unified Modeling Language.

בקוד ה C++ נוצרו מספר מחלקות (Class). בפרט,

#### 1. IStereoMatching

ממשק להפעלת אלגוריתם לחישוב התאמת סטראו. הממשק מונע את חשיפתו של המימוש הפנימי של מחלקת StereoMatching למחלקות תוכנה אחרות. במידה ונוסיף אלגוריתם סטראו חדש על האלגוריתם לממש את פונקציות הממשק.

#### 2. StereoMatching

מחלקה הכוללת את מימוש האלגוריתם. המחלקה מקבלת שתי תמונות ואובייקט פרמטרים ומחזירה תמונת עומקים. המחלקה יורשת מ IStereoMathcing.

#### 3. StereoMatchingFactory

מחלקה היוצרת אובייקט מסוג StereoMatching בעל פרמטרים שונים בהתאם לדרישות המשתמש.

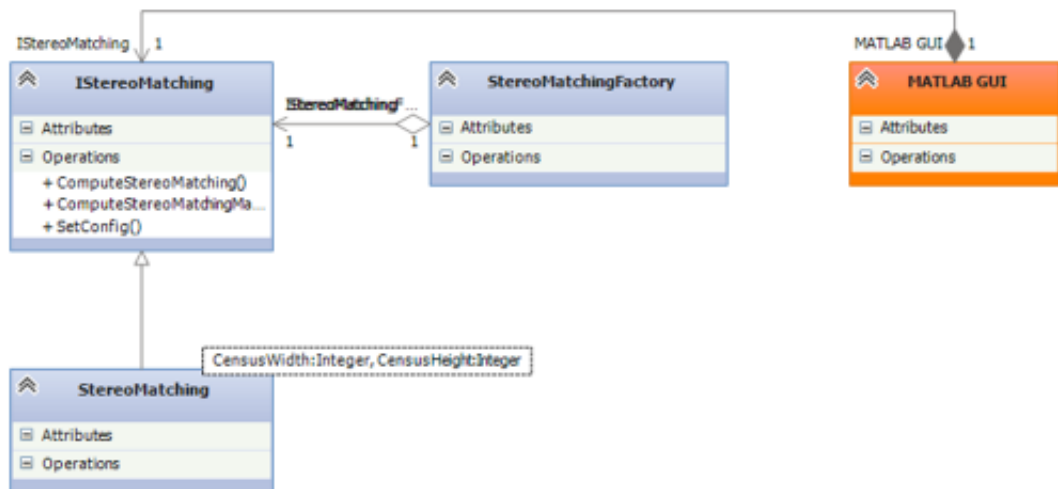
בקוד ה Matlab התבצע שימוש במחלקה אחת

#### 4. Matlab GUI

מחלקה הכוללת את ממשק המשתמש, לוגיקה לשינוי פרמטרים, טעינת תמונות, התממשקות מול קוד ה C++ והרצת האלגוריתם.



# UML - UNIFIED MODELING LANGUAGE



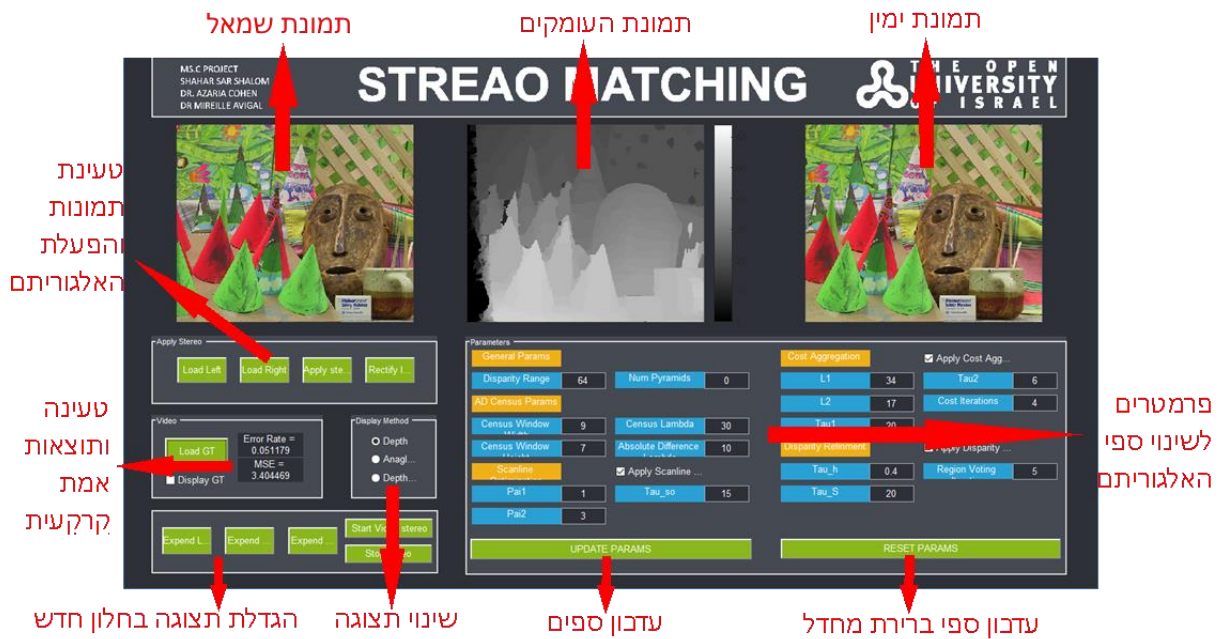
איור 25 - מבנה עיצוב התוכנה

6.4. ממשק משתמש

להלן ממשק המשתמש המשמש להפעלת האלגוריתם להתאמת סטראו, טעינת תמונות ושינוי הפרמטרים והספים משלבים א', ב', ג' ו-ד'.



איור 26 – ממשק משתמש (GUI) להפעלת האלגוריתם



איור 27 – פירוט החלקים השונים מאיור 26 - איור ממשק המשתמש

## 7. תוצאות ומסקנות

### 7.1 מבחני הערכת האלגוריתם

על מנת לאמוד אלגוריתם להתאמת סטראו יש לבחון אותו מול Ground Truth. אלגוריתם AD-Census שמומש בפרויקט נבחן מול מבחן הערכה בשם Middlebury Stereo Evaluation [13].

**מבחן הערכה** (Evaluation Dataset) כולל מספר תמונות סטראו **לאימון** (Training) ו**לבחינה** (Testing). תיקיית האימון כוללת מספר תמונות סטראו, כל תמונה כוללת תמונת עומקים המשמשת כ Ground Truth ותמונה עם אזורים מוסתרים שאינם מחושבים כחלק מתוצאת האלגוריתם. תיקיית הבחינה כוללת מספר תמונות סטראו בלבד, את תוצאות הבחינה ניתן לשלוח לאתר Middlebury ע"מ להעריך את ביצועי האלגוריתם, ובמידה וביצועי האלגוריתם טובים ניתן להתווסף לטבלה המדרגת את האלגוריתמים השונים. על מנת להעריך את אלגוריתם AD-Census שמומש בפרויקט נתייחס לתיקיית האימון של מבחן הערכה היות והיא כוללת תמונות Ground Truth.

מבחן הערכה Middlebury כולל שתי גרסות. מבחן הערכה Version 2 הכולל 4 תמונות סטראו בגודל ממוצע של 400x300 פיקסלים. מבחן הערכה Version 3 הכולל 15 תמונות סטראו בגודל ממוצע של 700x500 פיקסלים.

להלן תוצאות האלגוריתם AD-Census להתאמת סטראו אשר נבחן מול תמונות ה Ground Truth שמתוכן חושבו כמות הפיקסלים השגויים. כמקובל במבחני **התאמת סטראו צפופה** (Dense Stereo Matching) ערכו של פיקסל הוגדר כשגוי אם ההפרש בין ערך ה Disparity שהתקבל מהאלגוריתם לבין ערך ה Ground Truth גדול מפיקסל אחד, נגדיר מדד זה בשם **Pixel Error Rate – PER**.

$$f(i, j) = \begin{cases} 1, & \text{if } (D_{AD-Census}(i, j) - D_{Ground Truth}(i, j)) \geq 1 \\ 0, & \text{Otherwise} \end{cases}$$

$$PER = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j)$$


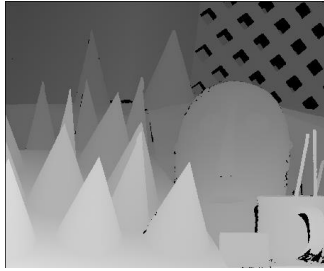
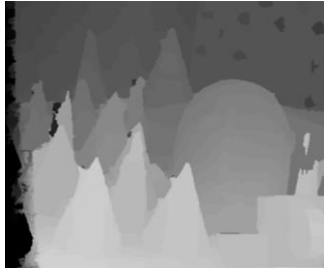

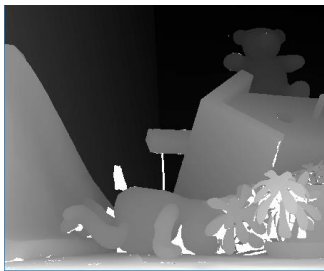
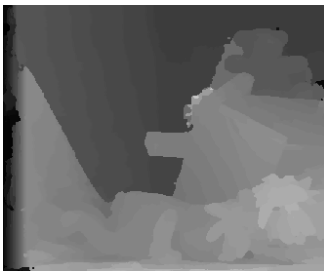
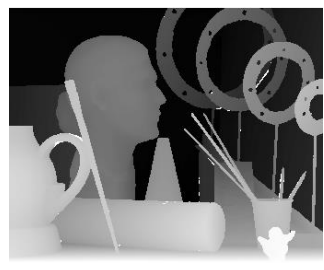


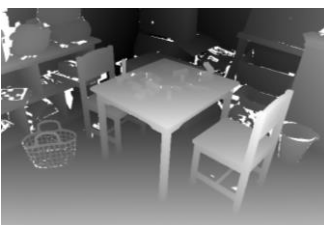
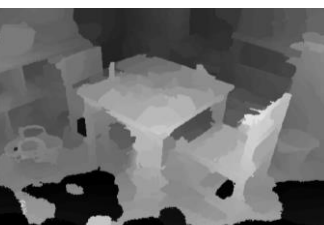
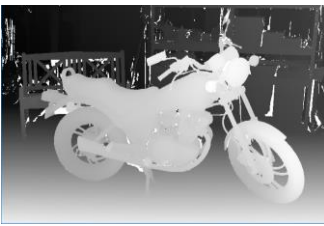
נקרא ל  $D_{Ground Truth}(i, j)$  ערך ה Disparity המתקבל מתמונת ה Ground Truth עבור פיקסל  $(i, j)$ , נקרא ל  $D_{AD-Census}(i, j)$  ערך ה Disparity המתקבל באמצעות אלגוריתם להתאמת סטראו AD-Census עבור פיקסל  $(i, j)$ .  
מדד נוסף הוא **שגיאה ריבועית ממוצעת**, Mean Square Error – MSE.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (D_{AD-Census}(i, j) - D_{Ground Truth}(i, j))^2$$

תוצאות האלגוריתם מוצגות בטבלה 1, בעמודה השמאלית מופיעה תמונת סטראו שמאל הנכנסת לאלגוריתם, בעמודה האמצעית ניתן לראות את ה Ground Truth ובעמודה הימנית ניתן לראות את תוצאות האלגוריתם AD-Census כפי שמומש בפרויקט. בנוסף בטבלה 2 ניתן לראות את שם התמונה, מקור התמונה, ממדי התמונה, טווח החיפוש  $nDisp$  בפיקסלים, מדדי הביצועים PER ו MSE וזמן ריצה.

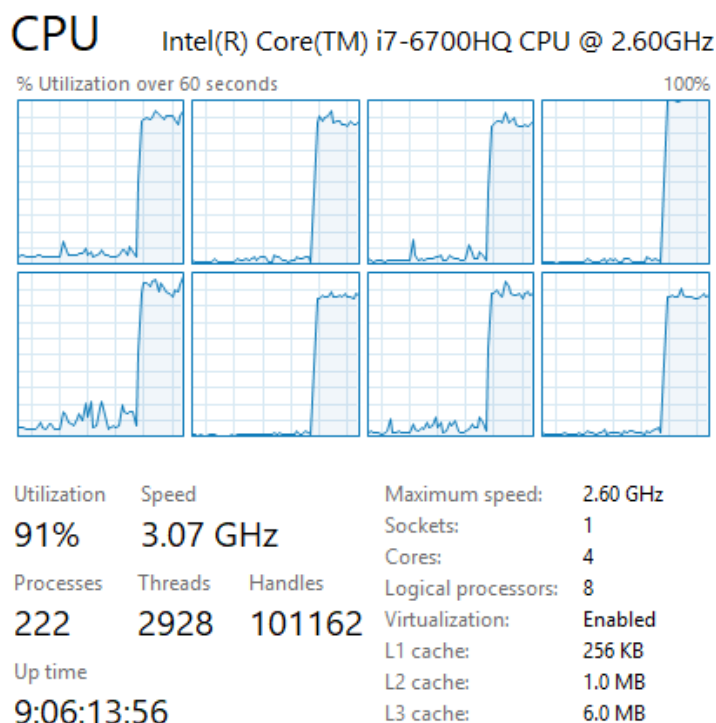
טבלה 1 – תוצאות האלגוריתם AD-Census עבור תמונות נבחרות

תמונה מספר	תמונות כניסה	Ground Truth	אלגוריתם AD-Census

1			
2			
3			
4			
5			

תמונה מספר	שם התמונה	מקור התמונה	גודל תמונה	nDisp	MSE	PER (%)	זמן ריצה (ms)
1	Cones	Version 2	450x375	59	3.4	5.1	6372
2	Teddy	Version 2	450x375	59	4.12	7.5	6401
3	Art	Version 3	347x277	64	15	23	3971
4	Playtable	Version 3	600x463	73	308	49	22771
5	Motorcycle	Version 3	741x497	70	10.86	21	23856

האלגוריתם רץ במחשב Intel i7-6700HQ, 3.50GHz. עבור תמונת סטראו בגודל 384x288, זמן הריצה היה 3562ms והוצגה נצילות מעבדים של 90 אחוז. באיור 28 ניתן לראות את ביצועי המעבדים ונצילותם במהלך הרצת האלגוריתם.



איור 28 – ביצועי המעבדים ונצילותם במהלך הרצת האלגוריתם

תוצאות האלגוריתם משביעות רצון וטובות בעיני המתבונן והן מול תוצאות ה Ground Truth. נציין שהאלגוריתם המקורי שפורסם ב 2011 נבחן על Version 2 והפיק תמונת עומקים עם ביצועי  $PER = 3.97\%$  בממוצע. האלגוריתם AD-Census שמומש בפרויקט זה נבחן ב Version 2 על 3 מתוך 4 התמונות (אין Ground Truth זמינה עבור התמונה הרביעית) והפיק תמונת עומקים עם ביצועי  $PER = 4.5\%$  בממוצע, ערך הקרוב לאלגוריתם המקורי. ייתכן וההבדל בין התוצאות נובע ממחסור בתמונת בדיקה וטיפול שונה בפיקסלים השגויים בשלב ד' של האלגוריתם. כעת נבחן את ההשפעה של כל שלב באלגוריתם על התוצאות. ביצענו ניסוי כך שבכל שלב בניסוי הופעלו רק חלק משלבי של האלגוריתם, בטבלה 4 מוצגים הביצועים של כל ניסוי בבחינת Middlebury Version 2. כאשר  $\sqrt{\quad}$  מסמן אם שלב זה הורץ באלגוריתם ו X מסמן אם שלב זה לא הורץ באלגוריתם.

טבלה 4 - תוצאות מבחני הערכה Middlebury Version 2

שלב א' – AD – Census Cost function	שלב ב' Cross Based Cost Aggregation	שלב ג' Semi Global Matching	שלב ד' Multistep Disparity Refinement	PER (%)
√	X	X	X	40.9
√	√	X	X	5
√	X	√	X	8
√	√	√	X	4.8
√	X	X	√	21.1
√	√	X	√	4.9
√	X	√	√	4.5
√	√	√	√	4.69

האלגוריתם שמומש בפרויקט זה נבחן במבחן הערכה Middlebury Version 3 וביצועיו היו  $PER = 29\%$ . כעת נבחן את ההשפעה של כל שלב באלגוריתם על התוצאות. ביצענו ניסוי כך שבכל שלב בניסוי הופעלו חלק משלבי של האלגוריתם, בטבלה 5 מוצגים הביצועים של כל ניסוי בבחינת Middlebury Version 3.

טבלה 5 - תוצאות מבחני הערכה Middlebury Version 3

שלב א' - AD Census Cost function	שלב ב' Cross Based Cost Aggregation	שלב ג' Semi Global Matching	שלב ד' Multistep Disparity Refinement	PER (%)
√	X	X	X	56.2
√	√	X	X	27.2
√	X	√	X	26.5
√	√	√	X	28.1
√	X	X	√	44
√	√	X	√	28.3
√	X	√	√	23.8
√	√	√	√	29.3

בניגוד למצופה לא כל אחד משלבי האלגוריתם בהכרח משפר את התוצאות, התוצאות הטובות ביותר התקבלו כאשר הופעלו שלבים א', ג' וד' באלגוריתם, עם  $PER = 4.5\%$  עבור Version 2 ו  $PER = 23\%$  עבור Version 3.

קיים חסרון לאלגוריתם בהיותו רגיש לבחירת פרמטרים הן מבחינת תוצאות והן מבחינת זמני ריצה. נכון לכתיבת שורות אלו האלגוריתמים הטובים ביותר למציאת התאמת סטראו מבוססי אלגוריתמי למידה. אלגוריתמים אלו יחקרו במסגרת עבודת המשך - עבודה מסכמת.

## 8. סיכום

בעבודה זו, הוצג והוסבר אלגוריתם AD Census [5] להפקת מפת עומק מתמונה. האלגוריתם מומש ב C++ באופן יעיל תוך ניצול מספר מעבדים הקיים על ידי מיקבול תהליכים. באמצעות Graphical User Interface של Matlab נכתב ממשק משתמש נוח להרצת ובדיקת האלגוריתם.

המימוש נבחן מול מדדים מקובלים והודגמו תוצאות משביעות רצון ודומות למימוש המקורי.



- [1] O. Faugeras, Three-dimensional computer vision: a geometric viewpoint, The MIT press, 1993.
- [2] Szeliski, Richard, Computer vision, algorithms and applications, Springer, 2010.
- [3] Wikipedia, "Image rectification," [Online]. Available: [https://en.wikipedia.org/wiki/Image\\_rectification](https://en.wikipedia.org/wiki/Image_rectification).
- [4] Wikipedia, "Image rectification," [Online]. Available: [https://en.wikipedia.org/wiki/Image\\_rectification](https://en.wikipedia.org/wiki/Image_rectification).
- [5] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Want and X. Zhang, "On Building an Accurate Stereo Matching System on Graphic Hardware," *IEEE*, pp. 467-474, 2011.
- [6] Nvidia, "CUDA zone," [Online]. Available: <https://developer.nvidia.com/cuda-zone>.
- [7] Wikipedia, "Hamming distance," [Online]. Available: [https://en.wikipedia.org/wiki/Hamming\\_distance](https://en.wikipedia.org/wiki/Hamming_distance).
- [8] wikipedia, "Summed-area table," [Online]. Available: [https://en.wikipedia.org/wiki/Summed-area\\_table](https://en.wikipedia.org/wiki/Summed-area_table).
- [9] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222-1239, 2001.
- [10] G. Van MeerbergenM. VergauwenM. PollefeysL. Van Gool, "A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming," *International Journal of Computer Vision*, vol. 47, no. 1-3, p. 275–285, April 2002.

- [11] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 328-341, 2008.
- [12] Wikipedia, "Parallel computing," [Online]. Available: [https://en.wikipedia.org/wiki/Parallel\\_computing](https://en.wikipedia.org/wiki/Parallel_computing).
- [13] R. Szeliski and D. Scharstein, "Middlebury Stereo Evaluation," [Online]. Available: <http://vision.middlebury.edu/stereo/>.
- [14] K. Zhang, J. Lu and G. Lafuit, "Cross-Based Local Stereo Matching Using Orthogonal Integral Images," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1073-1079, 2009.
- [15] Z. Moratto, "Semi-Global Matching," 2013. [Online]. Available: <http://lunokhod.org/?p=1356>.
- [16] R. H. a. A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge: Cambridge University Press, March 2004.