

## דו"ח הגשת מטלה 2

### מגישים:

- שגיא גוילי, ת.ז. 203638804
- אור אשכנזי, ת.ז. 205711419

### תרגיל 1 – Epipolar Geometry

ויזואליזציה של צמדי התמונות הנדרשים:

#### Im\_family – 7Point Algorithm



מדדים:

The Algebraic distance is  $1.0467817089322905e-15$

The Epipolar distance is  $2.4788745840802253e-27$

#### Im\_family – 8Point Algorithm



מדדים:

The algebraic distance is  $0.011677134967515546$

The Epipolar distance is  $0.8310941729012347$

### Im\_court – 7Point Algorithm

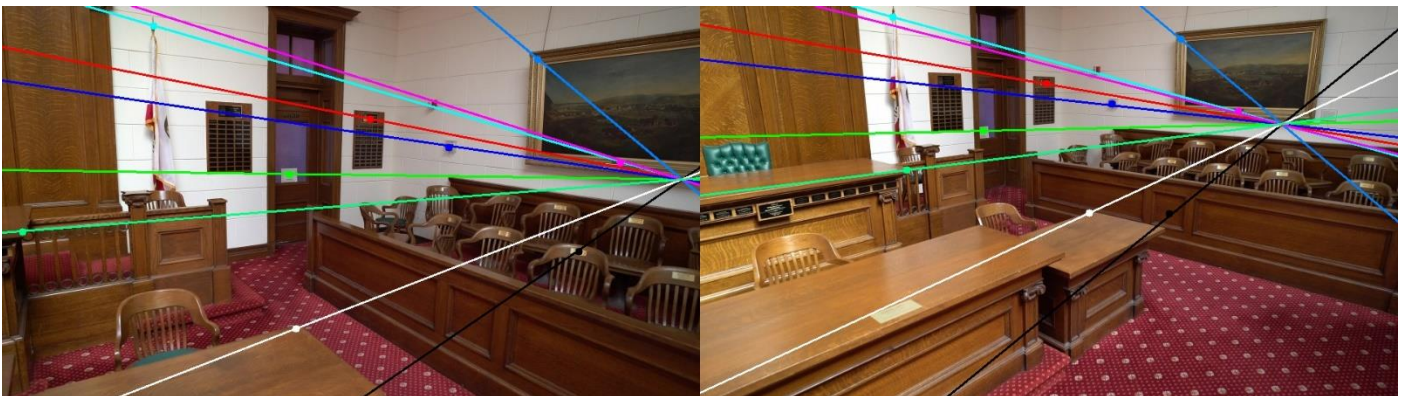


מדדים:

The algebraic distance is  $3.489272363107635e-16$

The Epipolar distance is  $8.44992892838244e-28$

### Im\_court – 8Point Algorithm




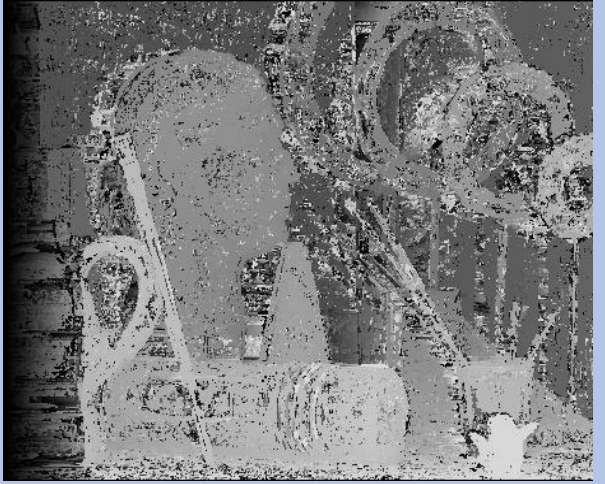
מדדים:

The algebraic distance is  $0.013948397690447745$


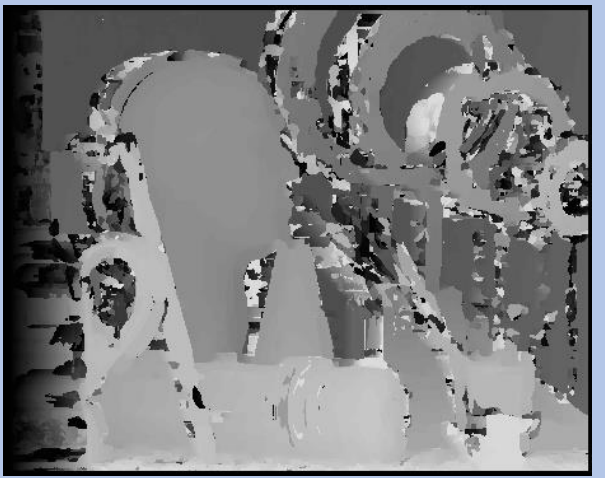
The Epipolar distance is  $4.527090053364976$





**Art:** Window Size = 3

SSD		NCC	
AvgErr	10.101034691439533		11.07408084190009
MedErr	2.333333333333332		1.333333333333321
Bad05	70.7553286805621%		61.56217579930209%
Bad4	42.40663019900028%		43.1263557483731%
image			

**Art:** Window Size = 9

SSD		NCC	
AvgErr	9.095381782122114		9.611280432274908
MedErr	1.0		0.6666666666666643
Bad05	64.46056705725213%		56.30744945662073%
Bad4	37.94790844514602%		36.155060409204054%
image			

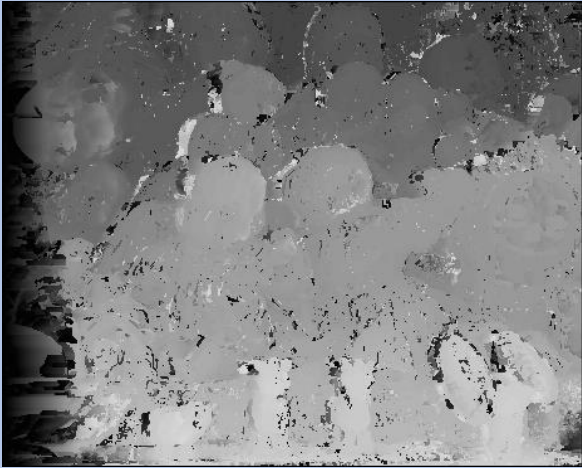
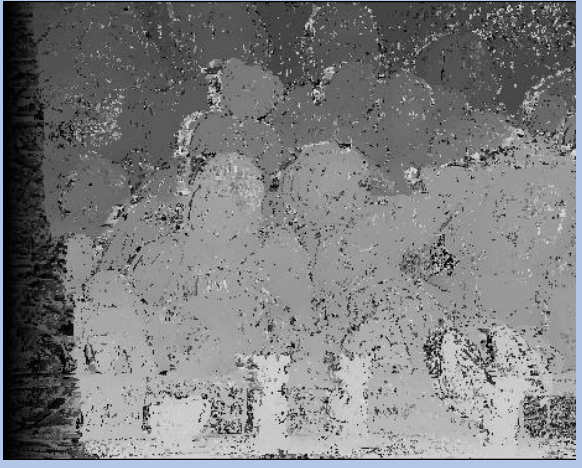
**Art:** Window Size = 15

SSD		NCC
AvgErr	9.439668259886721	10.265565176046646
MedErr	1.3333333333333357	1.0
Bad05	66.80388378669203%	62.121193163334254%
Bad4	40.367483296213805%	40.217962513450615%
image		

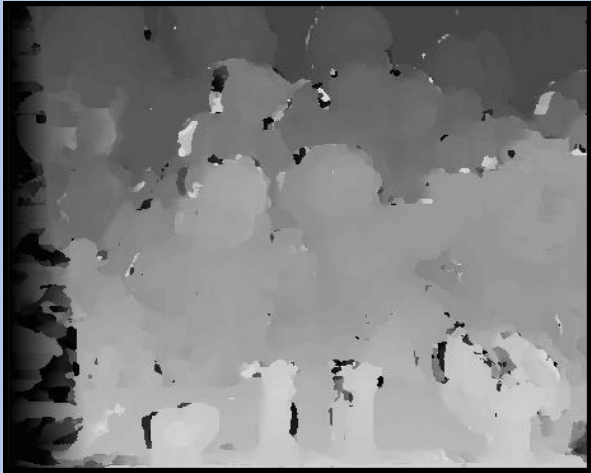
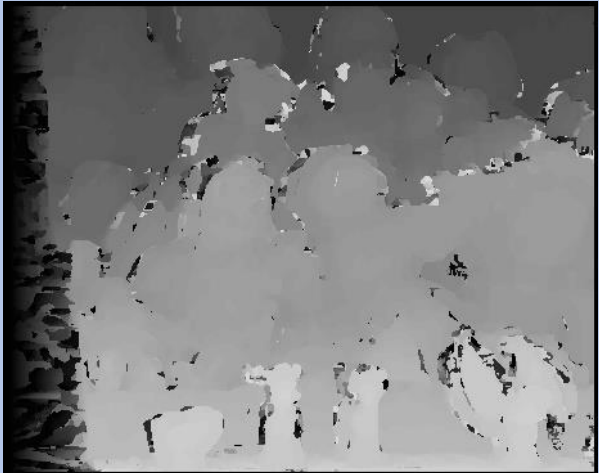
בעבור סט התמונות Art אנחנו מבחינים כי אנחנו מקבלים תמונות קוהרנטיות יותר כאשר אנחנו משתמשים במדד NCC מאשר מדד SSD, על מנת למצוא התאמה בין קואורדינטה בתמונה השמאלית לבין קואורדינטה בתמונה הימנית. ניתן לראות זאת על פי רוב המדדים, אך בעיקר על פי תמונות ה- disparities עצמן.

מבחינת חלונות החיפוש אנחנו נעדיף להשתמש בחלון חיפוש בו  $K = 9$ . עבור חלון חיפוש בו  $K = 3$  נקבל ארטיפקטים הנובעים מהיעדר אינפורמציה מספקת בתוך חלונות החיפוש, מה שמוביל לתוצאות שגויות במדדי ה-SSD ו-NCC. כאשר חלון החיפוש הוא בגודל  $K = 15$  נקבל את הבעיה ההפוכה – כעת בחלונות החיפוש יש יותר מדי אינפורמציה, דבר המוסיף אלמנטים מיותרים לחישוב ה-SSD / NCC ובהתאם לכך יותר התאמות שגויות.



**Dolls:** Window Size = 3

SSD		NCC	
AvgErr	6.303259297683047	7.622168647238202	
MedErr	0.66666666666666643	0.33333333333333357	
Bad05	55.56741016693388%	49.62156936716024%	
Bad4	24.17299349240781%	28.159483165141943%	
image			

**Dolls:** Window Size = 9

SSD		NCC	
AvgErr	4.939228543095947	5.3434136765628475	
MedErr	0.66666666666666643	0.33333333333333357	
Bad05	50.60227065751928%	45.69060773480663%	
Bad4	17.899945358508894%	18.215651751563353%	
image			

**Dolls:** Window Size = 15

	SSD	NCC
AvgErr	4.838100481302604	5.319511523735642
MedErr	0.66666666666666643	0.6666666666666664
Bad05	54.80093090763495%	52.62255699306824%
Bad4	18.77580641125097%	19.228748029328596%
image		

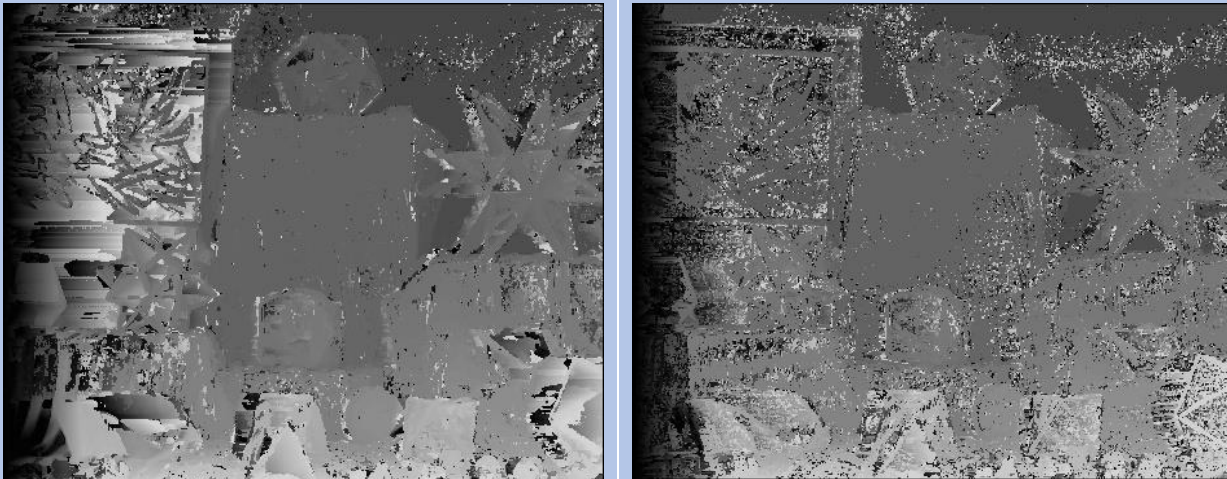
בעבור סט התמונות Dolls נבחין כי לא ניתן להכריע באופן חד משמעי איזה מדד הביא לנו את התוצאה הטובה ביותר, אם כי ניתנת עדיפות במעט לטובת מדד ה-SSD.

מבחינת חלונות ההתאמה אנחנו נשים לב כי עבור חלון בגודל  $K = 3$  אנחנו נקבל את ה-disparities של האובייקטים עצמם כמעט כמו בתמונת ה-ground truth, שכן התמונה מכילה בעיקר אובייקטים קטנים, לכן חלון חיפוש קטן יתאים עבורם. הבעיה מתעוררת כאשר עבור חלון מגודל זה נקבל תמונה מורעשת.

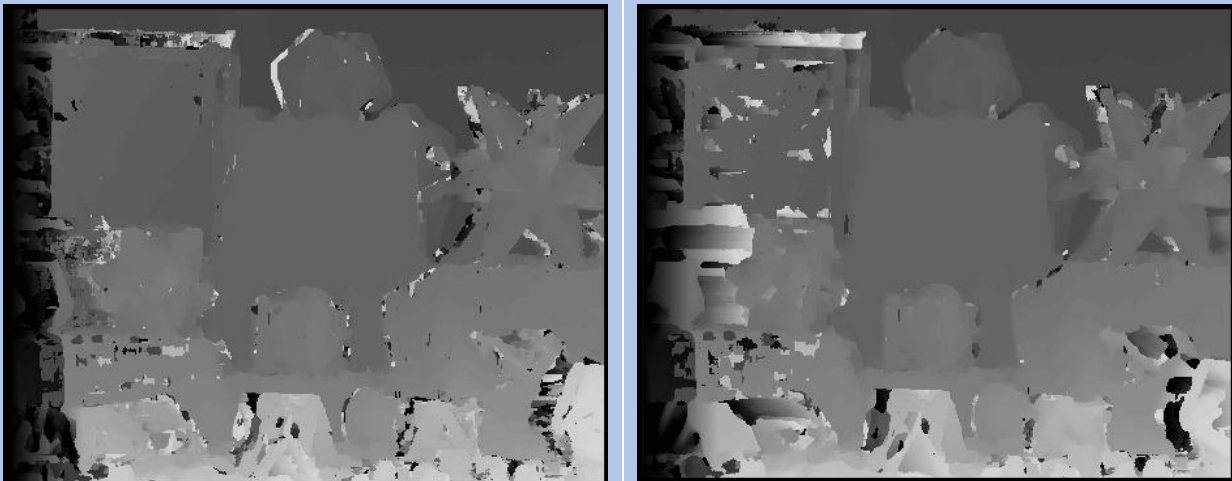
לעומת זאת עבור חלון מגודל 9 נקבל תמונה ללא רעשים, אך האובייקטים פחות ברורים בהם מפני שהם יותר נמרחים. על מנת להכריע נשתמש בארבעת המדדים הסטטיסטיים ונאמר כי עבור חלון מגודל  $K = 9$  נקבל את התוצאה הטובה ביותר.

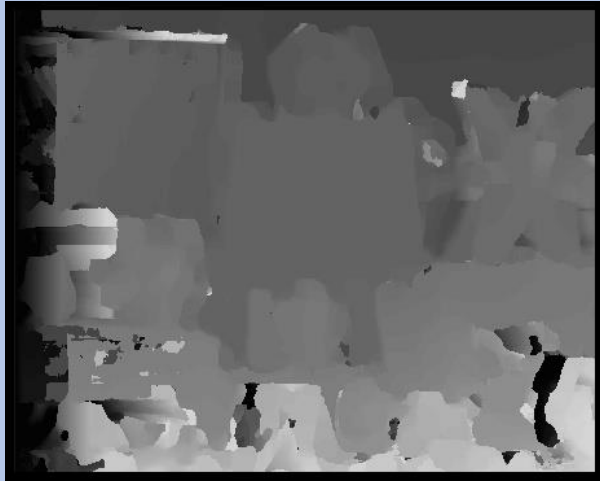
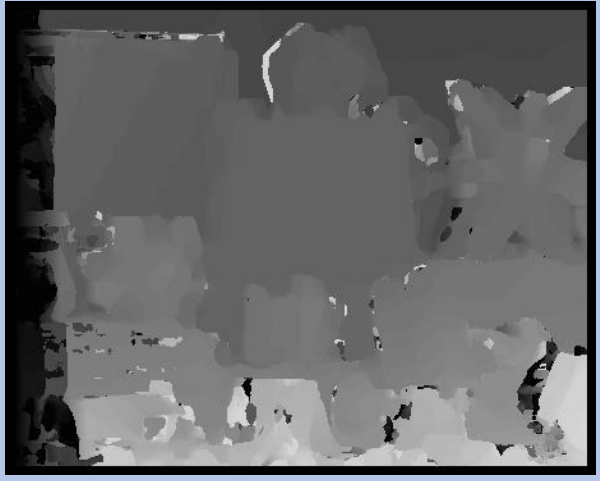


**Moebius:** Window Size = 3

SSD		NCC	
AvgErr	7.7662709453299374	7.928805526737715	
MedErr	0.6666666666666679	0.6666666666666643	
Bad05	56.19105441856078%	51.66698104310101%	
Bad4	35.100914835423936%	35.13628218428747%	
image			

**Moebius:** Window Size = 9

SSD		NCC	
AvgErr	5.453963531864084	4.980424179871693	
MedErr	0.3333333333333357	0.3333333333333357	
Bad05	45.04644526743974%	40.53427235747677%	
Bad4	24.741667172606398%	21.220933762370226%	
image			

	SSD	NCC
AvgErr	5.051433480977287	4.793477807529008
MedErr	0.33333333333333357	0.33333333333333357
Bad05	46.664247641450416%	43.90593328495283%
Bad4	24.764145041415382%	22.582643077000075%
image		

נבחין כי בעבור סט התמונות Moebius נקבל גם כאן תוצאות טובות יותר עבור בעבור מדד ה-NCC יותר ממדד ה-SSD. נבחין בכך שהתוצאות באות לידי ביטוי הן מבחינה ויזואלית והן מבחינת ארבעת המדדים הסטטיסטיים.

עבור חלון ההתאמה האופטימלי נסווג ונאמר שעבור שימור פרטים קטנים בתמונה עדיף חלון התאמה בגודל  $K = 9$ , אך עבור אובייקטים גדולים יותר בתמונה נקבל מעט יותר ארטיפקטים. מאידך, עבור חלון התאמה מגודל  $K = 15$  נקבל כי הוא דווקא כן משמר את האובייקטים הגדולים יותר, אך יוצר ארטיפקטים עבור אובייקטים קטנים. על מנת לפתור את סוגיה זו נתבסס על נתונים אשר נמדדו על ידי ארבעת המדדים הסטטיסטיים, ונבחר בחלון חיפוש מגודל  $K = 9$ . נציין כי עבור חלון התאמה מגודל  $K = 3$  ניתן להבחין כי אובייקטים קטנים אכן נשמרים, אך אין כמעט קירוב לאובייקטים גדולים, ובנוסף לכך גם ישנו את אותו רעש מלח – פלפל לאורך התמונה.

#### אבחנות בנוגע לזמני ריצה:

- יצירת תמונות disparities באמצעות מדד SSD רץ מהר יותר ממדד ה-NCC.
- דבר נוסף שאליו שמנו לב במהלך ההרצות הוא שככל ש- $K$  גדל, כך זמן הריצה קטן ולהיפך.



1. מצורף בתיקיית ההגשה סט התמונות בתיקיות הרלוונטיות + סרטון וידיאו אשר מריץ את התמונות.
2. ניתן להבחין על סמך רצפי התמונות, כי החלק של ההזזה נותן מידע והמחשה על המבנה התלת מימדי של הסצנה. מבחינה ויזואלית אנחנו תחילה יכולים לשים לב כאשר אנחנו מזיזים את המצלמה לאורך ציר ה- $z$  בסצנה, אז אנחנו רואים שמתקיים ה- $dolly$  effect מפני שהמצלמה היא פרספקטיבית. כלומר ככל שקירבנו את המצלמה אל התמונה אז האובייקטים בתמונה גדלו, ואילו כאשר הרחקנו את המצלמה מהתמונה האובייקטים קטנו. מה שיצא את האפקט הזה הוא שב- $pinhole$  camera קרני האור מתנקזים אל נקודה אחת, מה שמאפיין את המצלמה הפרספקטיבית (לעומת אורתוגרפית שבה כלל קרני האור מקבילים, מה שגורם לציר ה- $z$  כלל לא להיות רלוונטי ולא נותן לנו אפקט של תלת מימדיות).

היבט נוסף ואף חשוב יותר כך שההזזה של המצלמה נותנת לנו את המבנה התלת מימדי של התמונה הוא כאשר בכל תנועה (על כל ציר), נוצרים לנו חורים גדולים (blobs) שנראים כמו הצל של האובייקטים. כלומר ה- $blobs$  האלה נוצרים מפני שלא היה לנו מידע מה קיים מאחורי אותם אובייקטים ולכן נוצרו החורים. בגלל אותם  $blobs$  אפשר להסיק אילו

אובייקטים קרובים יותר אל המצלמה, ואילו רחוקים יותר. לצורך העניין נסתכל על התמונה המצורפת (תמונה im של 287 של ambush\_6).

אנחנו יכולים לראות ב-frame הזה שנוצרים  $blobs$  שלמעשה נראים כמו צלליות של הדמויות, אלא שבעקבות הזזת המצלמה ימינה בציר ה- $x$ , הדמויות עצמן זזו שמאלה, מה שלכאורה היה אמור לחשוף את אט פרטים נוספים בנוגע למה נמצא מאחוריהן באותו הצוק, אלא שהמידע הזה חסר לנו ולא ניתן להשלים אותו. לכן אנחנו יכולים להסיק שהדמויות קרובות יותר אלינו מאשר תחתית הצוק אשר נמצאת מאחוריהן.

החלק של הסיבוב לא תרם בהבנת המבנה התלת מימדי של התמונה, מפני שלא ניתן להבחין ולו ב- $blob$  hole אחד מתוך שלל תמונות הסיבוב. כל זאת משום שאפקט הסיבוב לא יוצר את ה- $dolly$  effect שצוין קודם לכן, ואין שינוי בצורת קרני האור אשר מגיעים אל ה- $pinhole$  camera.

### 3. בהמשך לסעיף הקודם:

- א. כפי שצינו קודם לכן ישנם חורים בתמונה שנוצרים בצורת  $blobs$  שהמקור שלהם נוצר מכך שאנחנו מזיזים את המצלמה במרחב, ובהתאם לכך האובייקטים שב-frame המקורי עברו טרנספורמציה הופכית אך כעת אנו אמורים לראות פרטים חדשים בתמונה החדשה שנוצרה, אך אותם פרטים חסרים לנו בגלל חוסר אינפורמציה על הסצנה. לכן נוצרות צלליות של האובייקטים שהיו יותר קרובים למצלמה. לעומת זאת סוג אחר של חורים שלא דנו בו עד כה, הם חורים בצורה של קווים. אותם קווים נוצרים מפני שלאחר הטרנספורמציה של הנקודות בתלת מימד (המכפלה של ה-extrinsic בכל קואורדינטה) ישנו סט של קואורדינטות שהתמפה לאותה הנקודה בתמונה החדשה. באמצעות אלגוריתם ה- $z$ -buffer שמימשנו באמצעות מפת העומקים הנתונה ידענו להכריע איזה ערך ניתן לפיקסל בהתאם לקואורדינטה שלה העומק המינימלי. כל קואורדינטה אחרת שמופתה אל אותה נקודה חדשה מתוך אותו סט של קואורדינטות לא תוצג, מלבד זו בעלת העומק המינימלי. כמו כן, כמובן שישנן קואורדינטות שלאחר הטרנספורמציה התמפו לקואורדינטות חדשות שחורגות מגבולות התמונה, ולכן גם הן לא יוצגו. חורים אלו בדמות קווים דקים מתרחשים בעקבות מיפוי בשיטת forward mapping ולא Inverse mapping. מה שאנחנו מבצעים בקוד שלנו הוא forward mapping מפני שאין לנו מספיק data על מנת לבצע inverse mapping.

ב. מבין שני סוגי החורים שדנו בהם ניתן לסווג כי החורים בצורת קווים ניתנים לפיתרון ומאידך החורים המשמעותיים יותר (ה-blob) אינהרנטיים. כפי שהסברנו עד כה החורים הגדולים האלו לא ניתן לפיתרון מפני שאין לנו אינפורמציה על המתרחש מאחורי האובייקטים בסצנה התלת מימדית שיצרנו, ולכן לא ניתן לגשר על הפער הזה. לעומת זאת, את החורים בדמות קווים כן ניתן לפתור ונציע בעבור כך את הפיתרון הבא: **אינטרפולציה** על הנקודות החסרות. ניתן להפעיל אינטרפולציית nearest neighbor / bilateral / bicubic כך שאותם פיקסלים בהם יש חורים נשתמש בצבע שיש לפיקסל השכן הקרוב ביותר שאינו חור. (הערה: הרעיון של Inverse mapping לא ניתן ליישום, שכן על מנת לעשות זאת אנחנו נצטרך תמונה נוספת של המרחב מנקודת מבט אחרת, בנוסף לדאטה שהיה נתון לנו עד כה. במקרה כזה ניתן להפעיל SIFT+RANSAC בין זוג התמונות, למצוא את הטרנספורמציה התלת מימדית שמקשרת בין התמונות כך שנוכל לבצע inverse mapping ואז נוכל לסנתז תמונות חדשות שמסנתזות את תנועת המצלמה מהתמונה הראשון אל השנייה)