מבוא לקו ישר

הנחות:

1. צבע הצורה שחור רקע לבן
2. אני מניח שהפיקסל השמאלי התחתון זה הראשית והתמונה נמצאת ברביע הראשון- אם לא נעשה את השינויים המתאימים.
3. עבור קו עם קיטוע "מספיק קטן" נתייחס כקו אחד
4. **הקדמה**

אנחנו מקבלים תמונה שנייצג בעזרת מטריצה שבכל תא מופיע 1 או 0, 1 מסמל שחור ו-0 מסמל לבן.

המטריצה מייצגת מישור, תא מייצג פיקסל (או וקטור).

בקבלת המטריצה נרצה להעביר את כל המידע של התאים השחורים, לשם כך נספור את כמות הפיקסלים השחורים נסמן את המספר ב-N, נגדיר מערך דו מימדי בגודל 2 על N. המערך הוא סדרה של וקטורים השייכים ל- בכל וקטור נכניס את הקורדינטות של פיקסל מסויים. למשל נניח שפיקסל במיקום המאה במערך הפיקסלים השחורים יש לו קורדינטה x שערכה 300 וקור' y שערכה 57 אז אפשר לסמן

\*\*\*הערה אין חשיבות למיקום הפיקסלים במערך.

1. **הגדרת שגיאה**

נסמן ב- את המרחק של פיקסל ה-n מהקו הישר

שגיאה מסדר 1:

שגיאה מסדר 2:

שגיאה מסדר m:

שגיאת אינסוף:

הסבר לגבי השגיאות- בסדר 1 לכל הנקודות יש משקל פורפורציונלי למרחק שלהם, ככל שסדר השגיאה עולה משקל הנקודות המרוחקות עולה ביחס לנקודות הקרובות, בנוסף אם אז .

\*\*\*נראה לי שכדאי גם להוסיף תלות באורך הקו ואולי גם ברוחב הקו

1. **זיהוי קו ישר**

\*\*\*נצטרך קודם לקבוע מינימום לכמות הפיקסלים שהחל ממנה נתחיל בתהליך הזיהוי, כי אם יהיה לנו רק שני פיקסלים ברור כי תמיד יצרו קו ישר.

3 דרכים

1. זיהוי ע"י תמונה אחרת של קו ישר וחפיפה בניהם
2. יצירת משוואת קו ישר מבחירה של שני פיקסלים (או שני קבוצות של פיקסלים), ובדיקה אם שאר הפיקסלים מקיימים את המשוואה.
3. שגיאת הקו המינימלית

**דרך 2**

אפשרויות לבחירת הנקודות:

1)בחירת שני הנקודות המרוחקות ביותר

**דרך 3**

הרעיון שעומד מאחורי השיטה:

הרעיון הוא שבמקום לבחור איזשהו קו מסויים ואז לבדוק את השגיאה נחשב את כל השגיאות על כל הקווים האפשריים ונבחר את הקו עם השגיאה המינימלית, אחרי שנקבל שגיאה מינימלית נוכל לבדוק אם הצורה היא קו ישר או לא. איך נעשה את זה? אנחנו נגדיר משוואת קו ישר שיכול להיות כל ישר במרחב, עכשיו נוכל להגדיר פונקציה שמקבלת את משוואה הקו (נכון יותר את הערכים a, b ו-c שמגדירים קו ישר) ונותנת את ערך השגיאה. כדי למצוא את נקודת המינימום אנחנו גוזרים את הפונקציה ומשווים לאפס.

נגדיר משוואת קו ישר

כאשר a ו-b לא יכולים להיות שניהם אפס.

נבצע הזזה של כל הנקודות כך שמרכז המסה יהיה בראשית (עושים את זה כדי לפשט את האלגברה), כלומר:

ובאופן דומה

נשתמש בהגדרת השגיאה מסדר 2:

כאשר N זה מספר הביטים, והביטוי זה מרחק הנקודה מהישר.

קיבלנו פונקציה עם שלושה משתנים ואנחנו מעוניינים למצוא את המינימום של הפונקציה.

תנאי הכרחי לנק' מינ' הוא שכל הנגזרות החלקיות יתאפסו, כדי למצוא את המינימום נאפס את הביטוי כלומר נדרוש:

אבל ולכן

כדי לאפס את הביטוי נדרוש

ובאופן דומה נקבל את המשוואה:

נסמן:

ונקבל את המערכת משוואות עם שני הנעלמים:

נפריד ל-3 מקרים

1 : מבטיח ש-ולכן נוכל לחלק ב-b את המשוואה הראשונה וב-a את המשוואה השנייה ונקבל את אותה משוואה:

נבחר b=1 כי יש לנו חופש בחירה, ונקבל

נשים לב שאם אז.

נגדיר גודל מתמטי נוסף, שיפוע הקו יוגדר להיות במקרה זה:

כאשר זה עבורו השגיאה המינימלית

2 :

נקבל שני פתרונות או או (נזכור שיש לנו חופש בחירה).

כאן או לא מוגדר בהתאמה.

3 :

נקבל אין סוף פתרונות.

\*\* לרוב זה יתקבל עבור צורות סימטריות ולכן נוכל להניח שהצורה היא לא קו ישר.

כדי למצוא את השגיאה נציב את הפיתרון במשוואה (3.3) כש-. נשים לב שקיבלנו תמיד 2 פתרונות אחד עבור נק' מינ ואחד עבור נק' מקס', לכן ניקח את התוצאה הקטנה יותר.

כדי להחזיר את משוואת הקו למערכת צירים המקורית נציב

הערה לגבי הגדלים הבאים:

- אפשרי להבין אותו כסימטריה סביב הצירים הראשיים, כאשר ריכוז גבוה של נקודות ברביע הראשון

והשלישי יניב ערך חיובית, לעומת ריכוז גבוה ברביע השני והרביעי יניב ערך שלילי. ריכוז שווה של

של נקודות יניב ערכים קטנים.

- ניתן להבין אותו כסימטריה ביחס לאלכסונים שחוצים את מרכז הראשית

צעדים:

|  |
| --- |
| 1. קבלת תמונה. 2. מספור הביטים השחורים. 3. יצירת מערך דו-מימד בגודל 2 על מספר הביטים השחורים. 4. הכנסת ערכי הקורדינטות (x,y) של כל הפיקסלים השחורים למערך. 5. חישוב מרכז מסה. 6. להפחית מכל קורדינטה את מרכז המסה. (3.2), (3.1) 7. חישוב הגדלים (3.4) , (3.5). 8. בדיקת שלושת המקרים. 9. אם זה מקרה שלישי להחזיר תשובה שלילית + זיהוי צורה סימטרית. וסיום. 10. קבלת a1 ו-a2 לפי המקרה המתאים. 11. קבלת השגיאה (3.3) ובחירת השגיאה המינימלית. 12. אם השגיאה קטנה תחזיר תשובה חיובית 13. אם השגיאה לא קטנה תחזיר תשובה שלילית |

**4. סידור מחדש של מערך הפיקסלים**

אם זוהה קו ישר נרצה לסדר מחדש את מערך הפיקסלים כך שנוכל לזהות התנהגויות של הקו.

רשימת שיטות:

1. סידור לפי סדר עולה של רכיבי x או y.
2. סידור לפי מרחק מהקו
3. סידור לפי הליכה על הקו

**שיטה 1:**

השיטה הזאת בעיקר תשמש אותנו לזיהוי איזורים מקוטעים/מדוללים

נחלק לשני מקרים:

א אם מהמשוואה (3.7) מקיים , נמיין בסדר עולה לפי רכיב ה-x.

כלומר אם ו- מסמנים מיקום במערך, אז אם אז .

במקרה ש- כלומר שני פיקסלים עם אותו רכיב x, אז גם פה נפריד לשני מקרים:

1. אם אז עבור כל הפיקסלים שרכיבי ה-x שלהם שווים, נמיין בסדר עולה של רכיבי ה-y, כלומר אם ו- מסמנים מיקום במערך, אז אם ו- אז .
2. אם אז עבור כל הפיקסלים שרכיבי ה-x שלהם שווים, נמיין בסדר יורד של רכיבי ה-y, כלומר אם ו- מסמנים מיקום במערך, אז אם ו- אז .

נסמן את המערך שמסודר בשיטה הזאת כ-.

ב אם מקיים או לא מוגדר, נמיין בסדר עולה לפי רכיב ה-y.

כלומר אם ו- מסמנים מיקום במערך, אז אם אז .

במקרה ש- כלומר שני פיקסלים עם אותו רכיב y, אז גם פה נפריד לשני מקרים:

1. אם או לא מוגדר אז עבור כל הפיקסלים שרכיבי ה-y שלהם שווים, נמיין בסדר עולה של רכיבי ה-x, כלומר אם ו- מסמנים מיקום במערך, אז אם ו- אז .
2. אם אז עבור כל הפיקסלים שרכיבי ה-y שלהם שווים, נמיין בסדר יורד של רכיבי ה-x, כלומר אם ו- מסמנים מיקום במערך, אז אם ו- אז .

נסמן את המערך שמסודר בשיטה הזאת כ-.

**שיטה 2:**

השיטה הזאת תשמש אותנו בעיקר לזיהוי פיקסלים מרוחקים ולא רצויים. נסמן ב- את המערך.

נסדר את הפיקסלים בסדר עולה של המרחקים שלהם ממשוואת הקו.

כלומר אם ו- מסמנים מיקום במערך, אז אם אז .

עבור פיקסלים שהמרחקים שלהם שווים נסדר אקראית. (\*\*\*הסיכוי שכמות של כמה פיקסלים להיות בעלת מרחק שווה הוא אפסי)

יהיה כדאי להוסיף למערך עוד שורה שתציין את המרחק של כל פיקסל.

**שיטה 3:**

הפיקסלים יסודרו על פי האלגוריתם הבא:

אלגוריתם הליכה על קו:

* הליכה על הקו הישר- נקודת ההתחלה שלנו תיהיה חיתוך משוואת הקו עם אחד מקווי המסגרות, נניח המסגרת התחתונה.,
  + אם נקדם את הקורדינטה y ב-1 ונמצא את x המתאים במשוואת הקו ונקפוץ לשם. וכן הלאה
  + אם נקדם את הקורדינטה x ב-1 נמצא את y המתאים במשוואת הקו ונקפוץ לשם.
  + אם נחסר ב-1 את קורדינטה x ונמצא את y המתאים במשוואת הקו ונקפוץ לשם.
  + אם m לא מוגדר נקדם את קורדינטת y ב-1 ונקפוץ לשם.
  + אם m=0 ונניח שאנחנו מתחילים מהמסגרת השמאלית נקדם את קורדינטת x ב-1 ונקפוץ לשם.
* הליכה על הקו המאונך- בכל פעם שאנחנו מתקדמים על הקו הישר נבצע הליכה על הקו המאונך. כיוון שאנחנו מתחילים מהאמצע נלך פעם אחת לכיוון מסויים נחזור לנקודה ואז נלך בכיוון השני
  + אם נקדם את הקורדינטה y ב-1 ונמצא את x המתאים במשוואת הקו ונקפוץ לשם עד שנגיע למרחק ואז נחזור לנק' ההתחלה נבצע את אותו תהליך רק במקום לקדם נחסר.
  + אם נבצע אותו דבר רק עם החלפת תפקידי הקורדינאטות.
  + אם m=0 או לא מוגדר נלך בקוו ישר ימינה ושמאלה או למעלה ולמטה בהתאמה.

השיטה הזאת טובה למקרים שנרצה זיהוי של פיקסלים המנותקים מהקו הראשי.

מערכים שימושיים אחרים:

1. מערך עם שני שורות, שורה אחת מציין את המרחק מהקו ושורה שנייה מציינת את כמות הפיקסלים



**5. זיהוי וסינון פיקסלים לא רצויים**

התמונה משמאל מדגימה מצב שיכולים להסתנן פיקסלים לא רצויים, זה יכול לקרות כאשר מסה רצינית של פיקסלים מגדירה קו וכמות קטנה של פיקסלים ניצלו זאת כדי להכנס. נרצה לזהות ולהפריד אותם.

רשימת שיטות:

1. סינון לפי המרחק ביחס לעובי הקו
2. סינון לפי המרחק ביחס לממוצע המרחקים
3. סינון לפי נתק מהקו הראשי
4. סינון פיקסלים באיזור הקצוות

**שיטה 1:**

נניח שחישבנו את עובי הקו, נסמן את עובי הקו ב-d, נוכל לסנן את כל הפיקסלים שמרחקם ממשוואת הקו גדול מ-, כאשר את הגודל של f נצטרך להגדיר מראש.

**שיטה 2:**

בדומה לשיטה הראשונה רק שפה נסנן את כל הפיקסלים שמרחקם ממשוואת הקו גדול מ-, כאשר את הגודל של f נצטרך להגדיר מראש, ו- זה השגיאה שזה בעצם ממצוע המרחק של כל הפיקסלים ממשוואת הקו.

**שיטה 3:**

נוכל לסנן פיקסלים שמנותקים מהקו הראשי, בשביל זה נצטרך להגדיר מרחק ניתוק. נגדיר את מרחק הניתוק כמרחק המקסימלי שבו פיקסלים יכולים להיות מנותקים מהקו בלי שום פיקסלים אחרים שמחברים אותם.

הערות:

1. מרחק הניתוק יוגדר כמרחק מעובי הקו ולא ממשואת הקו הישר
2. את מרחק הניתוק נוכל להגדיר כתלות בצפיפות הקו שאותה נוכל לחשב בצורה הבאה: אורך הקו כפול עובי הקו לחלק בכמות הפיקסלים. כלומר אם הקו מדולל הייתי מצפה שמרחק הניתוק יגדל ואם הקו צפוף אז מרחק הניתוק יקטן
3. השוני בשיטה שלוש לשיטות הקודמות הוא שזיהינו שהפיקסלים בעצם מנותקים מהקו, לכן נוכל לחפש פיקסלים קרובים יותר לקו מאשר בשני השיטות הוקדמות.



**שיטה 4:**

השיטה פה מתייחסת למקרים כמו שמתואר באיור משמאל, מצב שבו הפיקסלים המיותרים לכאורה נמצאים בקצוות.

נוכל לזהות אותם בעזרת איזורי דילול או עובי דק ביחס לקו הראשי.

אני חושב שפה נצטרך להשתמש בכל השיטות או בשילוב שלהם.

**6. מציאת קיטועים**

לפעמים נקבל קו מקוטע ונרצה לזהות את הקיטוע.

רשימת שיטות:

**שיטה 1:**

**7. מציאת קצוות**

בהנחה שהקו זוהה כקו ישר.

**דרכי פעולה:**

1. מציאת פיקסלים קיצוניים ותיקון
2. הליכה על משוואת הקו

אני מניח שהפיקסל השמאלי התחתון זה הראשית והתמונה נמצאת ברביע הראשון- אם לא נעשה את השינויים המתאימים.

\*\*נגדיר את שיפוע הקו הישר להיות

אם m יחשב כלא מוגדר.

**שיטה 1**

מציאת ארבעת הפיקסלים הקיצוניים, אם השיפוע של הקו חיובי אז קצה אחד זה ממוצע של הפיקסלים העליון והימני ביותר ולהפך הקצה השני. אם השיפוע שלילי אז קצה אחד זה הממוצע של הפיקסלים העליון והשמאלי ביותר וההפך הקצה השני. אם השיפוע הוא אפס אז אפשר למצוא רק את הפיקסל הימני ביותר והפיקסל השמאלי ביותר ואות דבר אם השיפוע לא מוגדר כלומר b=0 אז הפיקסל העליון ביותר והפיקסל התחתון ביותר

**שיטה 2**

מציאת נק' חיתוך של הקו אם אחת המסגרות, הליכה על הקו עד מציאת הפיקסל הראשון, פה יש שני אפשרויות או המשך הליכה על הקו ושמירת הפיקסל האחרון עד היציאה מהמסגרת, או קפיצה לנקודת החיתוך האחרת של הקו עם המסגרת והליכה עד מציאת הפיקסל הראשון.

בשיטה זו יכולים להתחרש מספר בעיות:

1. שני קווים מקבילים שמשוואת הקו עוברת בניהם וכד'

שיפור השיטה:

1. כדי לחסוך בזמן נמצא מראש את את ארבעת הפיקסלים הקיצוניים והם ייצרו לנו את המסגרת
2. כדי להמנע מהבעיה הנקובה, בכל פעם שאנחנו על פיקסל לבן נבדוק את כל הפיקסלים שנמצאים עד מרחק , כלומר , על קו שיוצא אנכית מהנקודה.

נגדריר את משוואת הקו המאונך:

נחלק ל-3 מקרים:

1. אם אז נבחר ואז שיפוע הקו המאונך יהיה את `c נמצא בעזרת הצבת הקורדינטות של הנקודה שממנה יוצא האנך.
2. אם אז השיפוע `m יהיה לא מוגדר ואת `c נמצא בעזרת הקורדינטה שממנה יוצא האנך.
3. אם m לא מוגדר אז ואת `c נמצא בעזרת הקורדינטה שממנה יוצא האנך.

אלגוריתם הליכה על קו:

* הליכה על הקו הישר- כיוון שאנחנו מתחילים באחת הקצוות אז נתקדם רק בכיוון אחד. נניח שאנחנו מתחילים מהמסגרת התחתונה,
  + אם נקדם את הקורדינטה y ב-1 ונמצא את x המתאים במשוואת הקו ונקפוץ לשם. וכן הלאה
  + אם נקדם את הקורדינטה x ב-1 נמצא את y המתאים במשוואת הקו ונקפוץ לשם.
  + אם נחסר ב-1 את קורדינטה x ונמצא את y המתאים במשוואת הקו ונקפוץ לשם.
  + אם m לא מוגדר נקדם את קורדינטת y ב-1 ונקפוץ לשם.
  + אם m=0 ונניח שאנחנו מתחילים מהמסגרת השמאלית נקדם את קורדינטת x ב-1 ונקפוץ לשם.
* הליכה על הקו המאונך- כיוון שאנחנו מתחילים מהאמצע נלך פעם אחת לכיוון מסויים נחזור לנקודה ואז נלך בכיוון השני
  + אם נקדם את הקורדינטה y ב-1 ונמצא את x המתאים במשוואת הקו ונקפוץ לשם עד שנגיע למרחק ואז נחזור לנק' ההתחלה נבצע את אותו תהליך רק במקום לקדם נחסר.
  + אם נבצע אותו דבר רק עם החלפת תפקידי הקורדינאטות.
  + אם m=0 או לא מוגדר נלך בקוו ישר ימינה ושמאלה או למעלה ולמטה בהתאמה.

**שיטה 3:**

נמצא את הקצוות כמו בדרך 2 ונבצע הטלה על משוואת הקו.

**שיטה 4:**

\*\*השיטה היא שיטה שאם קצת שינויים יכולה להתאים לכל קו ולא רק לקווים ישרים.

נגדיר מעגל ברדיוס . נבצע ריצה על כל הפיקסלים השחורים, במעבר על כל פיקסל נמקם את מרכז המעגל על הפיקסל,



בתמונות משמאל יש מקרי קיצון שמדגישים את הבעיתיות בכל אחת משתי השיטות הראשונות.

\*\*\*נצטרך לשכלל את השיטות\*\*\*

**הערות מסכמות:**

1. למרות שהשיטה השנייה פחות מהירה לפעמים נרצה לחשב את עובי הקו ואז בכל מקרה נצטרך לבצע הליכה.
2. בשיטה הראשונה במציאת הפיקסלים הקיצוניים- במקרה שיש כמה פיקסלים נוכל לבצע ממוצע שלהם.

**8. מציאת אורך ועובי הקו**

**אורך:**

את האורך נוכל להגדיר כמרחק בין הקצוות.

**רוחב:**

כדי למצוא את הרוחב הממוצע נבצע הליכה על כל הקווים המאונכים למשוואת הקו הישר. כל הליכה על קו מאונך תתבצע עד מרחק ממשוואת הקו הישר לשני הכיוונים. במהלך ההליכה נחפש את שני הפיקסלים המרוחקים ביותר נחשב את המרחק בניהם ולבסוף נבצע ממוצע של כל המרחקים.

**9. סינון פיקסלים להקטנת השגיאה**

במקרה שהשגיאה מאד קטנה אבל עדיין לא מספיקה להחשב כקו ישר, יכול להיות שיהיה אפשרי לסנן כמות מסויימת של פיקסלים מבודדים כדי להקטין את השגיאה לפי שאחנו מתחילים בהקטנת הרזולוציה.

רשימת שיטות:

\*\*\* אני חושב שאם שינויים מתאימים דרך הפעולה יכול להיות דומה לסינון פיקסלים לא רצויים

**10. הקטנת רזולוציית העבודה**

הנושא הזה הוא מאד נרחב שדורש התייחסות בפני עצמו נתמקד פה בשיטת עבודה שמתאימה לקווים.

אם לא זיהינו קו ישר נשתמש בהקטנת רזולוציה, כלומר נתייחס לחלק קטן יותר של התמונה.

רשימת שיטות:

1. חציית המישור בנקודת מרכז המסה לאורך הקו הסימטרי.
2. חלוקת המישור ל-4 בעזרת 2 קווי הסימטריה.

**שיטה 1:**

נעשה החלפת משתנים כדי שמרכז המסה יהיה בראשית לפי משוואות (3.1) ו-(3.2).

נגדיר משוואת קו שעובר בראשית .

נרצה למצוא מערכת קור' חדשה שבה הפיקסלים יהיו סימטרים ביחס לצירים, לשם כך נבצע החלפת משתנים סיבובית כדי לשמור על מרכז המסה בראשית.

עבור סימטריה סביב הצירים נדרוש-

נגדיר ו- בדומה ל-(3.4) ו-(3.5).

ונקבל

א אם אז נציב או או .

ב אם לא אז נוכל לחלק ב- ונקבל

שינוי בתוכנית! נשים לב שמתקיים מהמשוואה (3.6) . לכן במקום לסובב את הצירים ב-כך שתתקיים סימטריה סביב הצירים. נבחר את ציר הסימטריה להיות הקו הישר שמצאנו בפרק 3 או הקו המאונך לו.

\*\*\* צריך לבדוק איזה מהתשובות טובה יותר, לא בטוח בזה אבל אני חושב שעדיף לבחור בקו המאונך לו. אפשרי אולי גם לשלב בין שניהם

נבחר את אחד הישרים אזי משוואת הישר שלנו הוא כאשר a הוא אחד מן הפתרונות של (3.6)

נמיין את הפיקסלים -

אם אז הנק' מתחת לישר

אם אז הנק' מעל לישר

אם אז הנק' על לישר

ניצור שני סדרות חדשות של פיקסלים אחת של נק' מעל הישר, השנייה מתחת ואת הנקודות על הישר נפזר שרירותי בין הקבוצות. ונחזיר את הנק' לקורדינאטות המקוריות.

נסמן את אוסף הנקודות מעל הישר כ- ומתחת כ- .

נבצע זיהוי קו על או על . אם זוהה קו נבצע המשכיות של הקו (בהמשך נפתח אלגוריתם), אם לא זוהה קו נעבור לקבוצה השנייה ונבצע זיהוי קו. אם זוהה נבצע המשכיות של הקו, אם לא זוהה נקטין את שני הקבוצות לרזולוציות קטנות יותר וכך הלאה.

\*\*\* נצטרך להגדיר רזולציית מינימום

**שיטה 2:**



נחלק את המישור לארבע בעזרת שני קווי הסימטריה.

קווי הסימטריה הם: אחד הוא משוואת הקו הישר עם השגיאה המינימלית שמצאנו, והשני הוא הקו האנכי לו בנקודת מרכז המסה.(תמונה [10.1]).

ממשוואת קווי הסימטריה הם:

כאשר זה מהמשוואה (3.6), שווה 0 או 1 (תלוי באיזה מהמקרים קיבלנו בפרק 3 שיטה 3) ו- כש- הם רכיבי מרכז המסה.

\*\*\* אם קיבלנו צורה סימטרית נצטרך לחשוב על דרך לבחור את הקווי הסימטריה.

חלוקת הפיקסלים תתבצע כך:

1. קבוצת הפיקסלים הראשונה הם כל הפיקסלים שמעל הקו הישר או עליו ומעל לקו האנכי. כלומר מקיימים:

תחום זה יקרא הרביע הראשון. ונסמן ב- את אוסף הפיקסלים שנמצאים בתחום הזה.

1. קבוצת הפיקסלים השנייה הם כל הפיקסלים שמעל הקו הישר ומתחת לקו האנכי או עליו. כלומר מקיימים:

תחום זה יקרא הרביע השני. ונסמן ב- את אוסף הפיקסלים שנמצאים בתחום הזה.

1. קבוצת הפיקסלים השלישית הם כל הפיקסלים שמתחת הקו הישר או עליו ומתחת לקו האנכי. כלומר מקיימים:

תחום זה יקרא הרביע השלישי. ונסמן ב- את אוסף הפיקסלים שנמצאים בתחום הזה.

1. קבוצת הפיקסלים הרביעית הם כל הפיקסלים שמתחת הקו הישר ומעל לקו האנכי או עליו. כלומר מקיימים:

תחום זה יקרא הרביע הרביעי. ונסמן ב- את אוסף הפיקסלים שנמצאים בתחום הזה.

כעת נבצע זיהוי קו על התחומים הבאים: .

בהנחה שזוהו קווים בחלק מהתחומים או בכולם, נבצע את שני השלבים הבאים:

1. נבדוק אם תחומים שונים שזוהו בהם קווים שונים יושבים על אותו קו, נעשה זאת כך:

נבחר קו מסויים, ונבדוק האם משוואות של קווים אחרים, (\*\*\*או כל הקווים או הקווים שעבורם הקו המסויים עובר דרך התחומים שלהם או בסמוך לתחומים שלהם), קרובים למשוואת הקו המסויים.( \*\*\*נצטרך פה להגדיר מה נחשב קווים קרובים)

אם הקווים קרובים נאחד בין הקווים. אחרי שאיחדנו בין הקוום נבדוק אם הם מחוברים או נפרדים, נוכל לעשות זאת בעזרת הקצוות.

1. נבחר קודם את הקו עם השגיאה הקטנה ביותר ונבצע המשכיות קו (פרק 11) ואז נעבור לקו עם השגיאה השנייה הכי קטנה וכך הלאה.

לאחר מכן נוציא מכל המערכים , את הפיקסלים המשוייכים לקווים. ונעבוד על התחומים שנשארו בהם פיקסלים. נעבוד על כל תחום מארבעת התחומים בנפרד. נבצע זיהוי קו לכל תחום אם זיהינו קווים נתחיל מהקו עם השגיאה הקטנה נעשה איחוד קווים והמשכיות קו.

שוב נוציא מהמערכים את כל הפיקסלים המשוייכים לקווים, נבצע הקטנת רזולוציה לכל תחום בנפרד ונחזור לתחילת השיטה.

נצטרך להגדיר רזולציית מינימום.

**11. המשכיות הקו הישר**

הרעיון של המשכיות הקו הוא לנסות לזהות האם הקו הישר נמשך גם לתחומים אחרים, לפני שאנחנו יורדים ברזולוציה.

**שיטה 1:**

בהנחה שבתחום מסויים מצאנו משוואת קו ישר, נרצה לבדוק אם הקו ממשיך מעבר לתחום.

נניח ש- הוא עובי הקו שחישבנו.

נסתכל על כל התחומים (התחומים שדרגת הרזולוציה שלהם שווה לזו של הקו הישר) שבהם משוואת הקו הישר עובר. נבחר את כל הפיקסלים שנמצאים במרחק ממשוואת הקו, כשאר הוא מקדם ביטחון שנצטרך להגדיר אותו. נסדר את כל הפיקסלים במערכים חדשים לפי השיטות השונות שמופיעות בפרק 4. ונבצע התאמה בין מערכים אלה לבין אלו של הקו הישר.

צריך לראות איך בדיוק נעשה את ההתאמות האלה.

במקרה שמצאנו התאמה נרצה לבצע סינון פיקסלים שלא שייכים.

אפשרות קו שנפספס בשיטה זו מופיעה בתמונה משמאל.

למרות כשנבצע תהליך זיהוי של קווים ננסה להמנע מצורות עם מילוי, כלומר נרצה לראות רק מסגרות. ובנוסף נוכל לזהות את הקו ברזולוציות אחרות, ולעשות איחוד קווים.

**12. דיאגרמת סיכום**