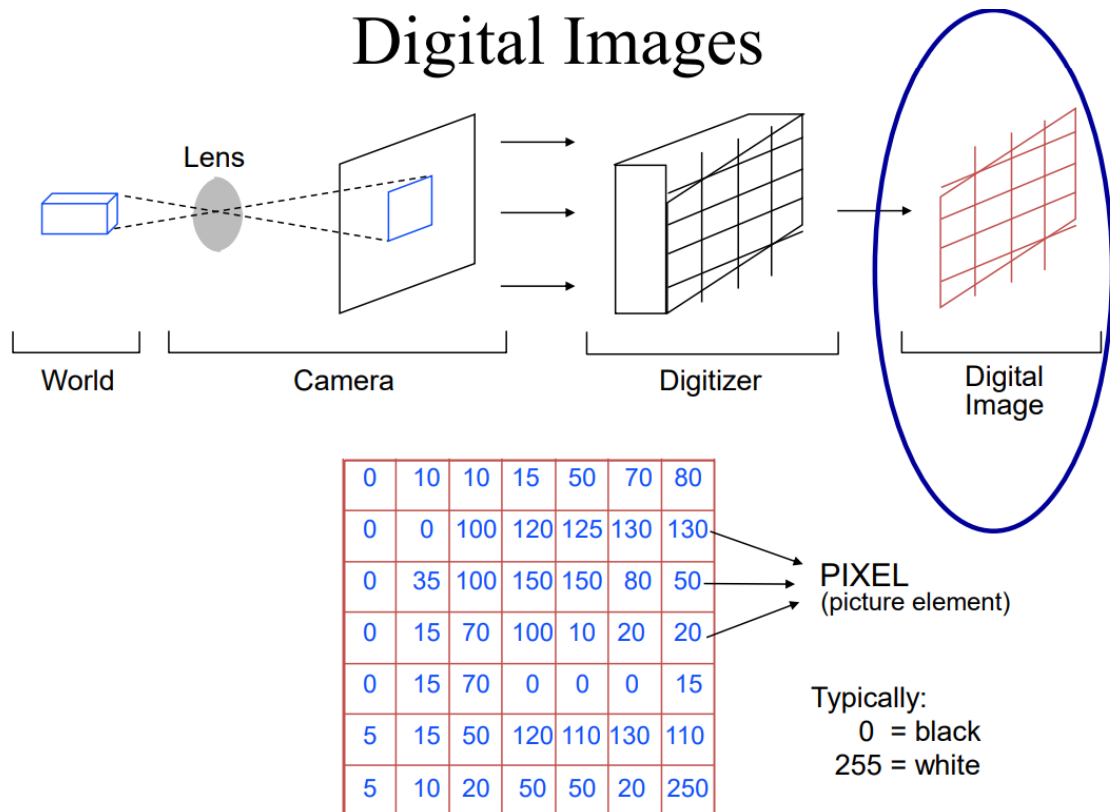


עיבוד תמונה

הרצאה 3



מעריך גלאים (digitizer) קולט את אותם פוטונים שבעצם מוחזרים מהאובייקט שאנחנו מצלמים והופך לתמונה

דיגיטלית

תמונה דיגיטלית היא מטריצה של כמה תאים

פיקסל זה תא עם ערך

על מנת שהתמונה תהיה יותר ברורה מקטינים את גודל הפיקסלים

לגבי תמונות במאטלב

אפשר לעשות import\export לכמה פורמטים של תמונות

הכי מקובלים הם JPEG רגילים לראות את זה כ-JPG

אפשר גם לשמור את התמונה בגדלים שונים בהתאם לצורך שלנו

סוגי תמונות

קיימים 4 סוגים בסיסיים:

תמונה בינארית – ערכים של הפיקסלים 0\1

לדוגמא דף מספר אותיות בשחור וכל שאר הלבן

Grayscale – ערכי אפור זאת אומרת מ0 עד 255 ערכי אפור (תמונות ישנות)

True color – אדום ירוק כחול, ערך בפיקסל יכול להיות 255 בחזקת 3

בערך 17 מיליון צבעים שונים אבל העין האנושית לא מסוגלת לראות את רובם

Indexed – נדבר עליו בהמשך

תמונה בינארית

תמונה בינארית היא מטריצה בינארית

יש 2 ערכים או אפסים או אחדות.

כל פיקסל הוא שחור או לבן

יתרונות => יעיל יותר מבחינת זיכרון

מספיקה במקרים רבים לזיהוי עצמים

רלוונטית לאפליקציות מסוימות בהן קל להפריד את העצמים מן הרקע

רקע יהיה בעל ערך 0 והוא הצבע הלבן

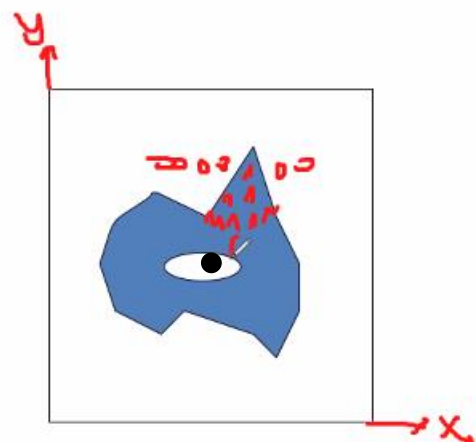
ופיקסלי האובייקט יהיו בעלי ערך 1 וזה הצבע השחור

מרכז כובד

יש לנו תמונה שבה יש אובייקט

הנקודה השחורה זה מרכז הכובד

$(x(m), y(m))$



אפשר לסכום את כל האיקסים שיש בהם אחד כלומר כמה אחדים כאלה ישלי

ממוצע עבור האיקסים וממוצע עבור העים
ואז מקבלים את המרכז של הכובד
(x(m), y(m))

$$x_m = \frac{1}{N} \sum_{i \in \text{object}} x_i$$

$$y_m = \frac{1}{N} \sum_{i \in \text{object}} y_i$$

תכונות גיאומטריות

חישוב מספר מאפיינים גיאומטריים עבור פיקסלים השייכים לאובייקט מסוים הינו חיוני לצורך זיהוי ומיקום האובייקט

גודל
על ידי הנוסחה אפשר לאפיין את הגודל של
האובייקט נקרא גם מומנט מסדר 0

$$A = \text{area} = \sum_{i=1}^n \sum_{j=1}^m B[i, j]$$

נניח אני רוצה לזהות שפיקסלים מסוימים שייכים לאובייקט מסוים

מומנטים

מומנט מסדר ראשון

d – מרחק ממרכז הכובד

p קבוצת פיקסלים

B(i) ערך הפונקציה בנקודה i

מומנט מסדר n:

$$M_n = \sum_{\forall i \in P} (d_i \cdot B_i)$$

לאפיין על ידי מומנט את האובייקט שלי

- קואורדינטות מיקום (מרכז כובד), מומנטים מסדר ראשון:

$$\bar{x} = \frac{\sum_{i=1}^n \sum_{j=1}^m j \cdot B[i, j]}{\sum_{i=1}^n \sum_{j=1}^m B[i, j]} = \frac{\sum_{i=1}^n \sum_{j=1}^m j \cdot B[i, j]}{A}$$

$$\bar{y} = \frac{\sum_{i=1}^n \sum_{j=1}^m i \cdot B[i, j]}{\sum_{i=1}^n \sum_{j=1}^m B[i, j]} = \frac{\sum_{i=1}^n \sum_{j=1}^m i \cdot B[i, j]}{A}$$

- כללית, מומנטים מסדר $q+p$:

$$m_{pq} = \frac{\sum_{i=1}^n \sum_{j=1}^m i^p \cdot j^q B[i, j]}{A}$$

אוריינטציה

רלוונטי עבור צורות אובייקטים

$$x^2 = \sum_{i=1}^n \sum_{j=1}^m r_{i,j}^2 B[i, j]$$

ניתן לאמוד על ידי התאמת ישר כלשהו לפיקסלי האובייקט $r(i,j)$ מבטא את השגיאה בין נקודת האובייקט לאותו ישר

B הערך של הנקודה (פיקסל)

קומפקטיות

נמדדת באמצעות:

$$\frac{P^2}{A} \geq 4 \cdot \pi$$

כאפשר P הוא היקף האובייקט A הוא השטח

במקרה קיצוני של קו ישר היחס הוא אינסוף

במקרה של המעגל היחס הוא 4 פאי כלומר המינימלי האפשרי

אם זה בצורת אליפסה אז היחס גדל

ייצוג תמונה בינארית

קידוד על ידי אורכי רצף

התחלה + אורכי רצף של פיקסלי 1 בכל שורה

אורכי רצף בכל שורה כשמתחילים עם 1

דרך אחת

ובסוגריים מסמנים מאיזה אינדקס מתחיל רצף האחדים וכמה יש ברצף

או שאפשר לכתוב כמה קבועים יש לנו כל פעם

דרך שנייה

נגיד בשורה הראשונה 3 3 2 3 בהתחלה

יש לנו קודם 3 אחדות ואז 3 אפסים ואז 2 אחדות

Binary image:

1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1
2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Start and length of 1 runs: 1 (1, 3) (7, 2) (12, 4) (17, 2) (20, 3)

2 (5, 13) (19, 4)

3 (1, 3) (17, 6)

Length of 1 and 0 runs: 1 3, 3, 2, 3, 4, 1, 2, 1, 3

2 0, 4, 13, 1, 4

3 3, 13, 6

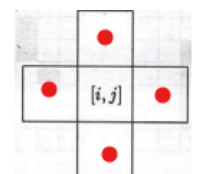
אלגוריתמים בינאריים

המטרה לקבץ יחד פיקסלי אובייקטים לתמונות אובייקטים

שכנות

שכנות בין פיקסלים מונח בתיאור גבולות העצמים וחלקי אזורים בתמונה

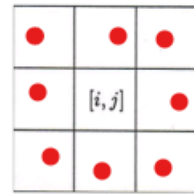
שכנות טיפוס 4



עבור פיקסל p במיקום (x, y) השכנים שלו יהיו השכנים האנכיים והאופקיים

במקומות: $(x+1,y)$ $(x-1,y)$ $(x,y+1)$, $(x,y-1)$

שכנות טיפוס 8



עבור כל פיקסל 8 השכנים הקרובים ביותר לפיקסל מגדירים את סביבת ה-8 שלו

התוספת פה אלה השכנים שנמצאים באלכסון שהם:

$(x+1,y+1)$, $(x-1,y-1)$, $(x-1,y+1)$, $(x+1,y-1)$

מסלול

סדרה של פיקסלים כך ש:

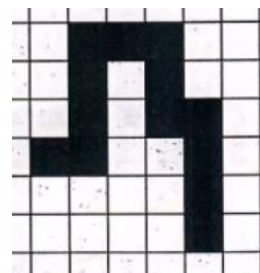
מסלול (path) סדרת פיקסלים $[i_1, j_1], [i_2, j_2], \dots, [i_n, j_n]$ כך ש- $[i_k, j_k]$ הינו שכן של $[i_{k+1}, j_{k+1}]$ עבור $0 \leq k \leq n-1$.

סט של פיקסלי 1 (האובייקט) יסומן על ידי s וייקרא גם foreground

קשירות

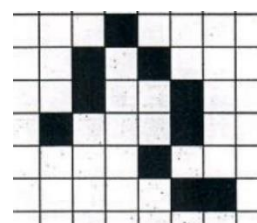
מסלול 4 הוא מסול שבו קיים בין שניים לאורך המסלול קשר מסוג 4

דוגמא ל-4 קשיר:



ואותו לדבר לגבי מסלול 8 רק עם קשר חלש יותר

דוגמא ל-8 קשיר



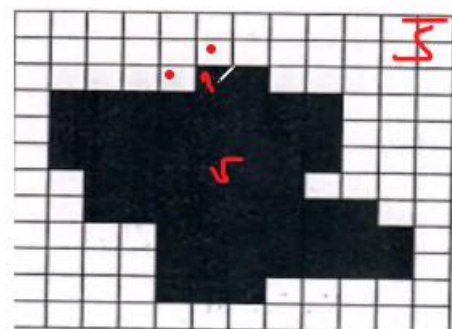
שתי נקודות Q ו- P השייכות לאזור S נקראות m קשירות אם ביניהם קיים מסלול m העובר באזור כלומר אם P שייך ל- S ו- Q שייך ל- S קיים מסלול מ- P ל- Q שמכיל אך ורק פיקסלים ששייכים ל- S .
 S זה אזור קשיר שבסוף כולם מתחברים
 קשירות מקיימת יחס שקילות
 אם כל הנקודות באזור הן m קשירות אז כל האזור נקרא m קשיר
 מרכיב קשיר $=$ סט של פיקסלים בו כל פיקסל קשור לשאר הפיקסלים הינו מרכיב קשיר
 רקע $=$ סט על המרכיבים הקשירים ב- S תג הכוללים פיקסלי קצוות של התמונה
 שאר המרכיבים של S תג נקראים חורים

בעצם לוקחים את התמונה
 ישלנו S שזה האובייקט הקשיר
 וישלנו לא אובייקט קשיר שזה יכול להיות רקע או בתוך האובייקט יכול להיות לי חור

פיקסלי גבול

הגבול של קבוצת פיקסלי S הכוללים את אותם פיקסלים ב- S שלהם יש שכנים מטיפוס 4 ב- S משלים (תג)
 את הגבול נסמן ב- S תג

נראה ש-1 הוא פיקסל גבול כי ישלו קשר מטיפוס 4 עם הרקע



(a) Original image

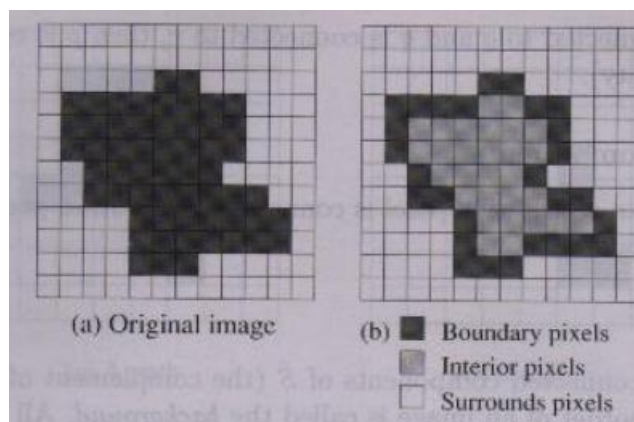
Surround pixels אלה שמקיפים את האובייקט

בתוך האובייקט עצמו

Interior פיקסלים פנימיים
 Boundary pixels פיקסלי גבול

גבול

גבול של אזור s הוא אוסף הנקודות של s שיש להן שכנים מחוץ ל- s
לדוגמא גבול 4 לפי שכנות מטיפוס 4:



קונטור וההיקף – אלגוריתם למציאת פיקסלי הגבול

1. מצא את נקודת ההתחלה $s \in S$ ע"י סריקה שיטתית של התמונה (שמאל ימין ומעלה מטה).
2. יהא פיקסל השפה הנוכחי c ; הצב $c = s$. נסמן ב- b את שכן-4 (ממערב) של s ;
($b \in \bar{S}$)
3. יהיו n_1, n_2, \dots, n_8 שמונת שכני-8 של c (עם כיוון השעון), באשר $n_1 = b$;
מצא את ה- i הראשון עבורו $n_i \in S$.
4. הצב $b = n_{i-1}, c = n_i$.
5. חזור על צעדים 3 ו-4, עד אשר $c = s$.

מתחיל נגיד מהשורה הראשונה בפינה השמאלית ועוברים על השורה
לא מצאנו פיקסל עוברים לשורה הבאה וכך הלאה עד שמוצאים את נקודת ההתחלה
(הפיקסל הראשון שנתקלים בו)

ברגע שמצאנו את הנקודה נסמן שכן 4 ממערב (כלומר מצד שמאל) ונסמן אותו ב- b
ומן הסתם b שייך ל- S משלים

כלומר מצאתי את הפיקסל הראשון של S וממערב לו

ואת הפיקסל עצמו נסמן כc

יהיו לנו n_1, n_2, \dots, n_8 כלומר 8 השכנים של c עם כיוון השעון וכאשר $b = n_1$
נבדוק עבור כל השכנים מי כן שייך לS (הבדיקה מתבצעת עם כיוון השעון)

ועבור אותו $n(i)$ הראשון ששייך לS נציב $c = n(i)$ ו $b = n(i-1)$

ועושים את זה שוב ושוב עד ש $c = s$ כלומר שסיימנו לעשות את כל הסיבוב

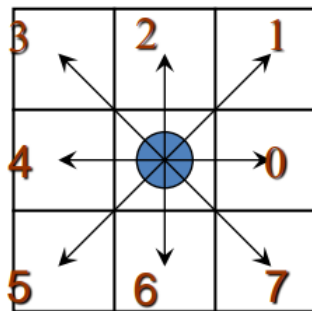
קוד שרשרת chain code

דרך אחרת לקידוד תמונה מורכבת מאזורים בעלי גוונים אחידים על ידי קידוד גבולות אזורים

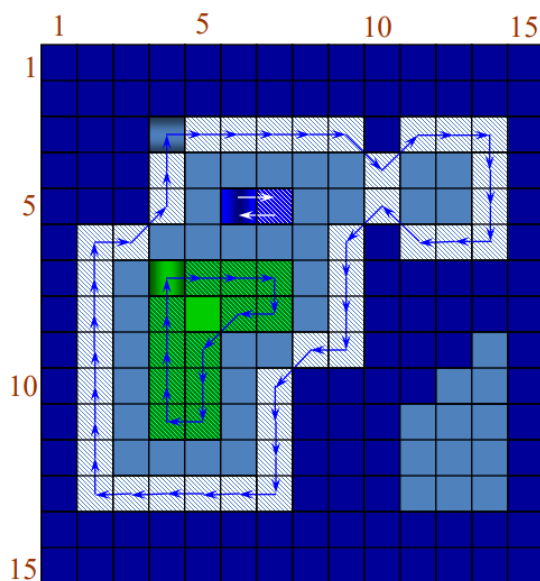
אז מה שיש פה: מיקום של פיקסל ההתחלה

גוון של האזור

שורה של מילות קידוד של הגבול



לדוגמא



● נקודת התחלה:

– (3,4)

● גוון:

– light blue

● קידוד הגבול:

0000071006664435666...

● נקודת התחלה:

– (7,4)

● גוון:

– green

● קידוד הגבול:

– 0006456642222

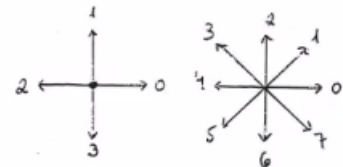
אז נגיד עבור הירוק

קודם חץ ימינה אז 0

ברגע שיורדים למטה זה 6 ואז שמאלה זה 4 וכך הלאה

ואז מגיעים לרצף של הקידוד

אפשר לעשות את זה עם שכנות מטיפוס 4 או שכנות מטיפוס 8



מבחינת יעילות עדיף שכנות 8 כי בשכנות מסוג 4 צריך יותר מספרים

קידודים וייצוגים של תמונות

לפעמים ייצוג כמטריצה דו ממדית הוא ייצוג בזבזני ואפשר לשמור את אותה תמונה באמצעות

הרבה פחות ערכים כי נרצה ייצוג קומפקטי ככל האפשר

אלגוריתם למציאת מרכיבים קשירים

המטרה למצוא את כל המרכיבים הקשירים

1. נעבור על התמונה בסריקת raster (משמאל לימין, שורה שורה)

2. רק אם זה פיקסל על אובייקט כלומר ערך 1

נבדוק אם אחד משכניו שויך לקבוצה והוא עליון או שמאלי בלבד אז נשייך את הפיקסל הנוכחי

לאותה קבוצה

נגיד



אם העליון והשמאלי באותה קבוצה אז גם הפיקסל האדום ישויך לקבוצה שלכם

אבל אם שני השכנים (העליון והשמאלי) שויכו לקבוצות שונות אז נאחד אותם לקבוצה משותפת

ונצרף את הפיקסל הנוכחי גם לקבוצה

ואם אין שכן משויך למעלה או משמאל אז עושים קבוצה חדשה

לדוגמא

רק אם זהו פיקסל על אובייקט (בעל ערך 1)

1. אם רק אחד מבין השכנים העליון והשמאלי (שכנות 4) שויך לקבוצה

שייך את הפיקסל הנוכחי לאותה קבוצה.

2. אם לשני השכנים מלמעלה ומשמאל

יש אותו שויך

שייך את הפיקסל הנוכחי לאותה קבוצה.

3. אם שני השכנים מלמעלה ומשמאל

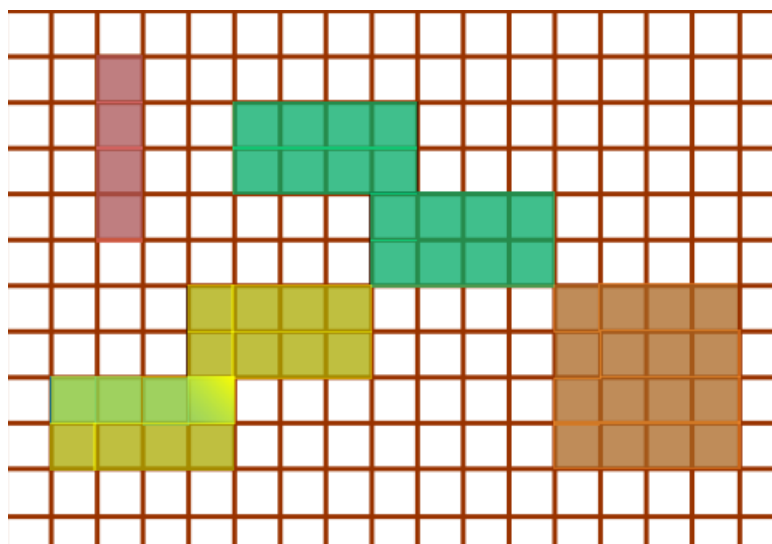
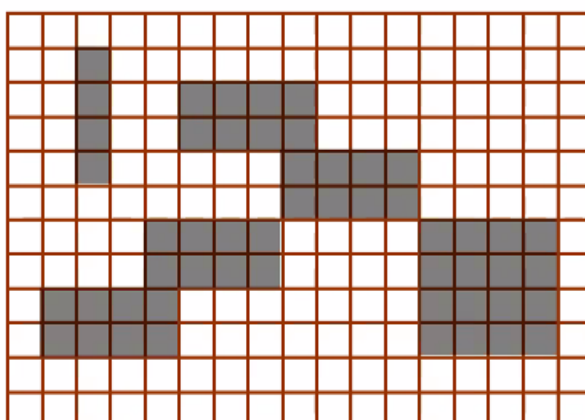
שוויכו לקבוצות שונות

צרף את שתי הקבוצות לקבוצה אחת,

ושייך את הפיקסל הנוכחי לאותה קבוצה.

● אם אין שכן משויך למעלה או משמאל,

המצא קבוצה חדשה ושייך את הפיקסל אליה.



מאפיינים טופולוגיים מספר Euler

פרמטר המשמש לאפיון צורה

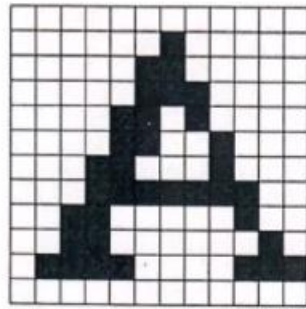
כאשר C הוא מספר המרכיבים H שווה למספר החורים

$$E \triangleq C - H$$

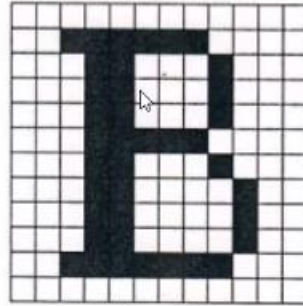
ישלנו מרכיב ובו יש חור אז מספר אוילר שווה למספר המרכיבים פחות מספר החורים

אז במקרה הזה יצא לנו $0 = 1 - 1$

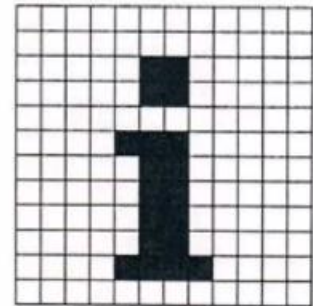
צורה אינה מתאפיינת בהכרח באופן חחע על ידי מספר אוילר



$E = 0$
מרכיב 1, חור 1



$E = -1$
מרכיב 1, חור 2

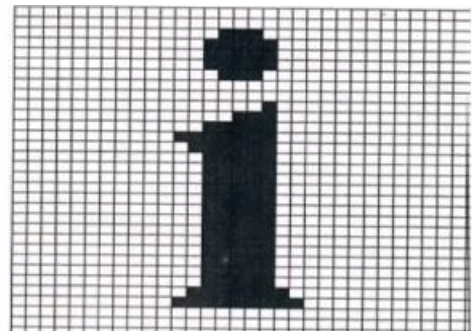
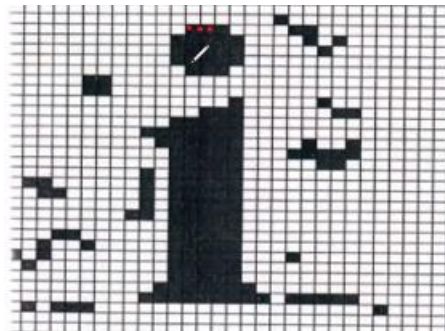


$E = 2$
מרכיב 2, חור 0

הוא עשוי לשמש בסיווג תווים בהקשר של OCR (זיהוי כתב וכאלה)
והוא רגיש לרעש

שימוש במסנן מידה size filter

על מנת למנוע רעשים אפשר להשתמש במסנן
בחרים T שיהיה גדול מספיק על מנת לנקות רעש אבל לא גבוה מידי שלא ימחק חלק מהתמונה
T יסמן איזה רכיבי פיקסלים צריך למחוק
לדוגמא עבור $T=10$ יסוננו רכיבים קשירים שגודלם פחות מ-10 פיקסלים



ואז אחרי סינון אפשר להשתמש באוילר
שיטה זו שיטה לסילוק רעש בהינתן תמונה בינארית
אז קודם נמצא אזורים קשירים ואחרי זה נמחק את כל האזורים הקשירים שגודלם מתחת לגודל
מסוים

עיבוד תמונה עושים פר מקרה כי אין איזה שיטה שטובה בשביל כל המקרים

מרחקים בין פיקסלים

בהינתן תמונה בינארית ניתן להגדיר מרחק בכמה אופנים

אך בכל מקרה חייבות להישמר התכונות של המטריקה

התכונות הן:

- 1) $d(p, q) \geq 0$ and $d(p, q) = 0 \iff p = q$
- 2) $d(p, q) = d(q, p)$
- 3) $d(p, r) \leq d(p, q) + d(q, r)$ אי שיוויזון המשולש

(d-מרחק)

המרחק הראשון נקרא גם מרחק אוקלידי

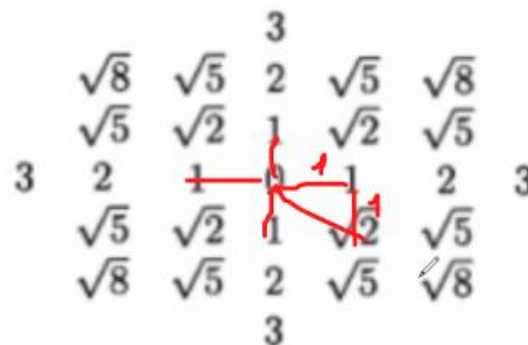
המרחק השני נקרא גם cityblock

המרחק השלישי נקרא גם chess

מרחק אוקלידי

נקרא גם מרחק קו אווירי

$$d([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$$



• לדוגמא:

$$\sqrt{1^2 + 1^2} = \sqrt{2}$$

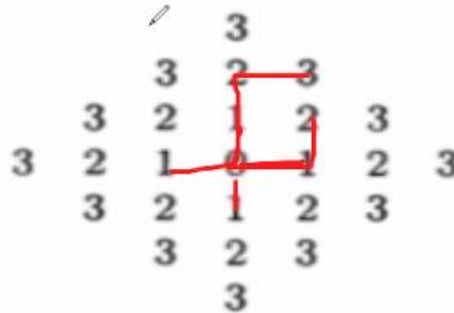
מרחק מנהטן cityblock

לא מרחק קו אווירי אלא כמו המרחק על פני רחובות ושדרות

העתק אופקי+העתק אנכי

$$d([i_1, j_1], [i_2, j_2]) = |i_1 - i_2| + |j_1 - j_2|$$

• דוגמא



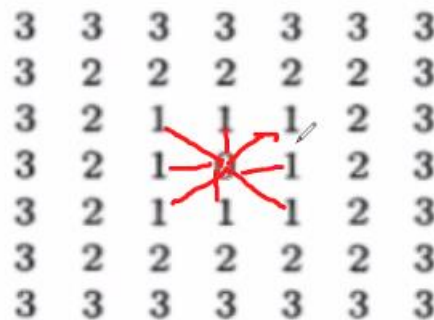
מרחק שחל chessboard

המקסימום בין ההעתק האופקי וההעתק האנכי

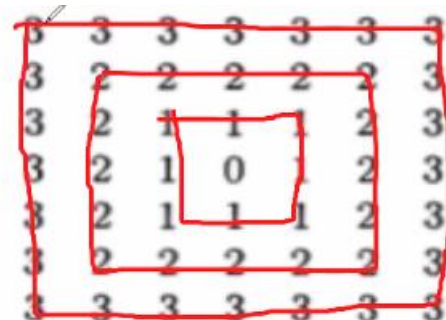
אפשר לנוע גם בצורה אלכסונית

$$d([i_1, j_1], [i_2, j_2]) = \max(|i_1 - i_2|, |j_1 - j_2|)$$

• דוגמא:



מרחק 1, מרחק 2, מרחק 3



בתמונה עדיף להשתמש בגישה האוקלידית כי אין שם הפרעה

טרנספורמציות מרחקים

המרחק בין פיקסלי אובייקט לרקע יכול לשמש במספר אפליקציות
בהינתן סט S נרחב לחשב עבור כל פיקסל ששייך ל S את המרחק של P לרקע S תג
חישוב המרחקים נעשה באופן איטרטיבי באופן הבא:

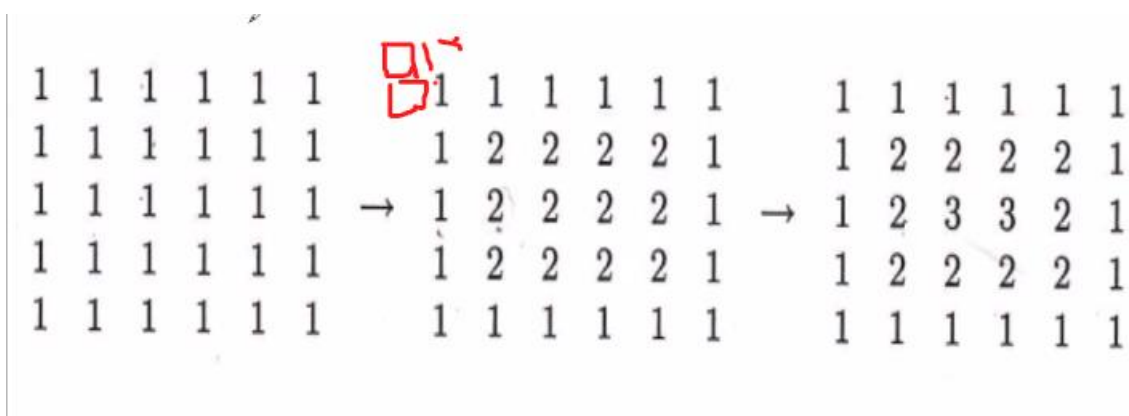
$$f^0[i, j] = f[i, j]$$

$$f^m[i, j] = f^0[i, j] + \min(f^{m-1}[u, v])$$

באשר m מס' האיטרציה עבור כל הפיקסלים $[u, v]$ כך ש- $d([u, v], [i, j]) = 1$
(הכוונה רק לשכני-4 של $[i, j]$).

למעשה רוצים לדעת כמה אנחנו רחוקים מהרקע

לדוגמא



שינוי ציר מדיאלי(אמצעי\מרכזי)

נועד לייצוג קומפקטי של אובייקטים
ונותן מענה לשאלות כמו האם פיקסל נמצא באזור כלשהו
סט פיקסלים S שהמרחק שלהם מ S תג מהווה מרחק מקסימלי לוקאלי קרוי $skelton \setminus MAT$
ה $skelton \setminus MAT$ יצוין על ידי S^* וזה מהווה ייצוג קומפקטי של S
כלומר ניתן לשחזר את S באמצעות S^* והמרחקים ל S תג של כל פיקסל ב S^*

$$d([i, j], \bar{S}) \geq d([u, v], \bar{S})$$

לכל הפיקסלים $[u, v]$ שבסביבת $[i, j]$ (שכנים מטיפוס 4) זה נותן קווים\נקודות שמייצגים את הצורה

לדוגמא

נרצה להמיר את כל פיקסלי 1 במרחקים שלהם לרקע

למה נרצה לעשות את זה כי זה יכול לתת אינדיקציה

בתמונה בינארית אפשר להגדיר עם איזה פונקציית מרחק עובדים ולהחליף

הציר האמצעי יהיה קבוצת הפיקסלים שהם מקסימום מקומי, הערכים הגדולים ביותר

כי זה הכי רחוק מהרקע

הציר האמצעי הוא ייצוג קומפקטי של התמונה ומאפשר מספר פעולות כמו הבחנה בין אזורים עגלגלים, מאורכים ומלבניים

grayscale תמונות גווני אפור

בעיקר תמונות רפואיות

בדרך כלל מ-0-255

256 רמות של אפור מספיקות לזיהוי רוב האובייקטים הטבעיים

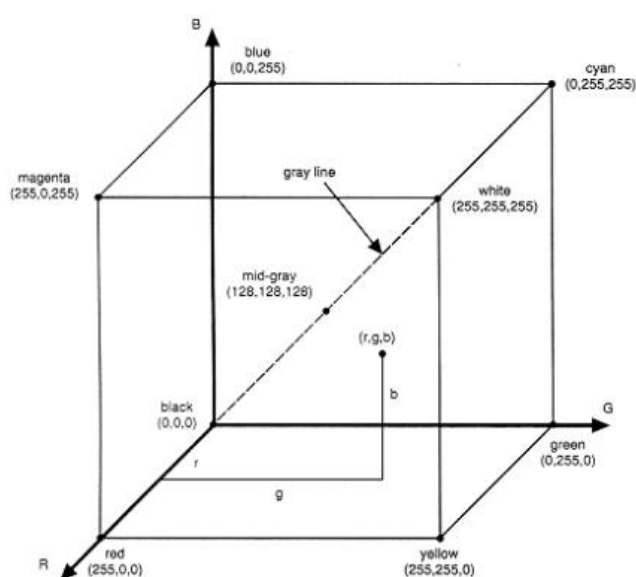
מיוצג על ידי 8 ביטים

תמונות צבע

בעיקר תמונות מהחיים

מסופק באמצעות ה-**RGB**

כל פיקסל משויך לווקטור תלת מימדי (r,g,b)



מודל נוסף רלוונטי הוא HIS והוא קרוב יותר לתפיסה האנושית Hue גוון כלומר הצבע הנתפס (אורך הגל הדומיננטי) Saturation רוויה, דילול על ידי הצבע הלבן (לדוגמא סגול בהיר סגול כהה) Intensity עוצמה, בהירות

קידוד שורות ו RLC

אפשרות אחת לקודד כל שורה לחוד

לכל קטע נרשום את צבעו ואורכו



זה יהיה קידוד חסכוני כאשר התמונה מורכבת מאזורים גדולים בעלי גוון אחד

עבור תמונות ברמות אפור כל גוון יקודד לפי ערכו

לפעמים משתמשים בשיטת קידוד RLC על ידי נקודת התחלה ואורך של האזור בצבע המסוים

בצורה הבאה:

G (1,1) 4 (2,1) 5 (2,13) 2 (3,11) 2
R (1,8) 5 (2,8) 5 (3,7) 4 (3,14) 1
B (1,5) 3 (1,13) 2 (2,6) 2 (3,1) 6 (3,13) 1