

## עיבוד תמונה – ת"ב מספר 1

### מגישים:

- ואדים חכם, ת"ר 311890156
- גלעד עיני, ת"ר 034744920

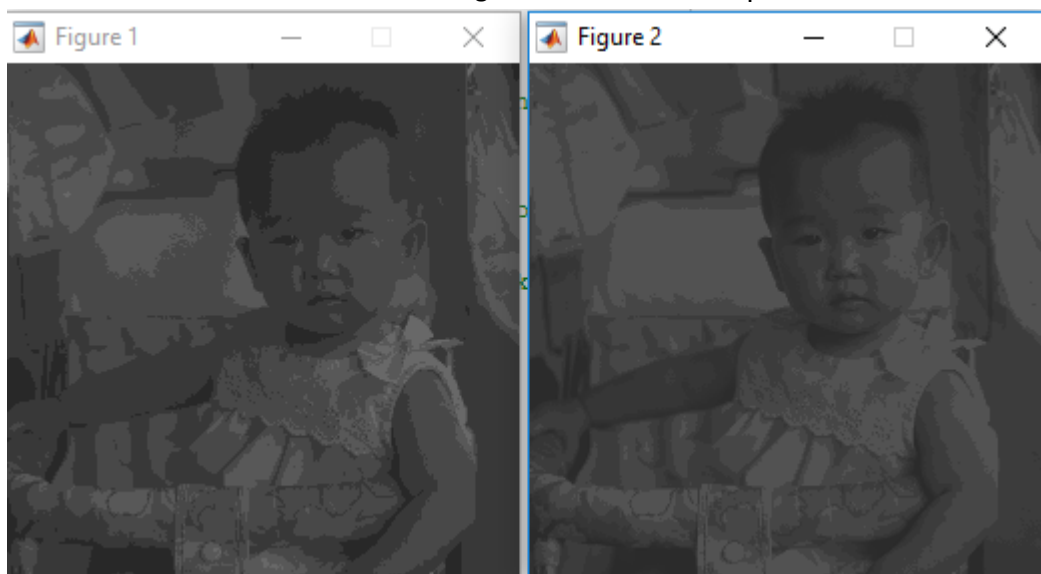
### תלויות של הפרויקט:

- שתי תמונות מה IMAGES של חגית שמצורפות להגשה על כל מקרה:
  - darkimage.tif
  - lakeScene.tif
- פונקציות מה TOOLBOX של חגית (לא מצורף):
  - readImage.m
  - showImage.m

### תשובות ותמונות לסקריפט 7

7א. השוואה בין כימות יוניפורמי לבין כימות אופטימלי ( $4=N$ )

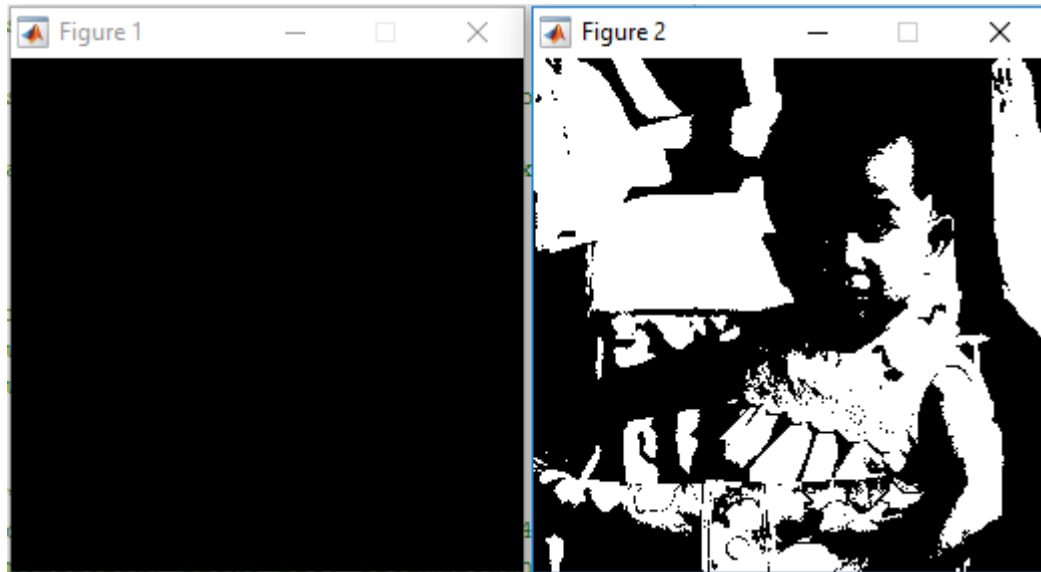
○ תמונה מקורית מה IMAGES : darkimage.tif



- ניתן לראות משמאל את הכימות היוניפורמי.
- מימין את הכימות האופטימלי
- האופטימלי יותר טוב.
- ביוניפורמי, היד הימנית נראית כחלק מהרקע. כמו כן, קשה להבדיל מתי המצח נגמר ומתי מתחיל השיער. כמו כן, רואים false contour בכל הרקע, במיוחד בפניה שמאלית עליונה.
- באופטימלי, רואים את המצח, השיער והיד הרבה יותר טוב.
- בחרנו בתמונה זו כי ידענו שהתמונה עברה השחרה(ע"י חגית), ושהיוניפורמי "יבזבז" צבעים(QI) סתם כי הוא יוניפורמי, להבדיל מהאופטימל שרוב הצבעים קרובים לתמונה.

7.ב. השוואה בין BINARIZE עם  $T=128$  ו-BINARIZEOPT (כלומר T אופטימלי)

○ תמונה מקורית מה-IMAGES : darkimage.tif



- ניתן לראות משמאל את `binarize('darkimage.tif',128)`.
- מימין את `binarizeOpt('darkimage.tif')`
- האופטימלי יותר טוב.
- לא יכלה להיות תמונה יותר טובה להמחיש את ההבדל.
- אנו מניחים, בלי להסתכל בערכי הצבעים של התמונה, שרובה אם לא כולה, ערכים מתחת ל-128. לכן, שקראנו עם  $T=128$ , יצאה תמונה שחורה.
- זהו יתרון ענק של ה-T אופטימלי, הוא יכול להתמודד עם תמונות בהירות ותמונות שחורות.

## ג. השוואה בין `binarizeOpt()` ל `binarizeOptAdaptive()`

○ תמונה מקורית מה `lakeScene.tif` : IMAGES



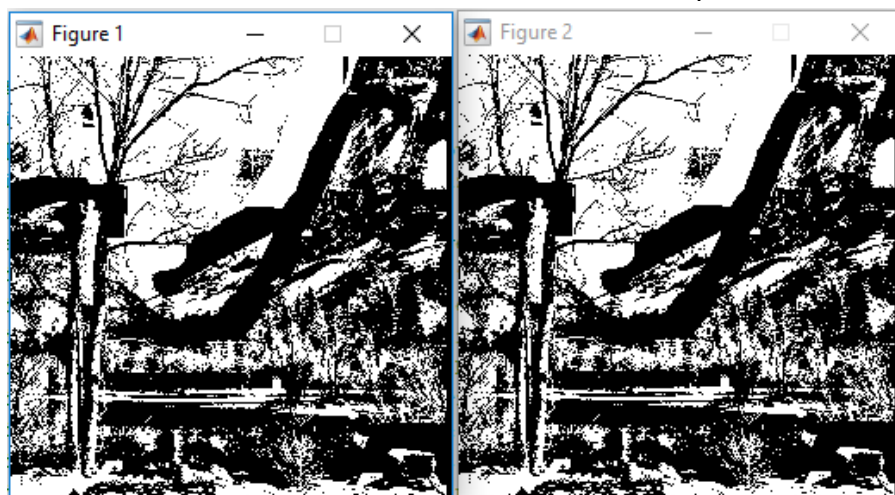
○ משמאל את `binarizeOpt('lakeScene.tif')`

○ מימין את `binarizeOptAdaptive('lakeScene.tif',31)`

- בחרנו בתמונה עם שינוי מהיר בפרטים כדי לקבל את המקסימום מהפונקציה.
- התמונה האדפטיבית יותר טובה מפני שקיבלנו הרבה יותר פרטים. ניתן לשים לב לכל הצמחיה מצד ימין למטה, שנמצאת לפני ההר. ניתן לשים לב גם לצמחיה הפזורה על המים. דבר נוסף, ניתן לשים לב למשהו משמעותי ביותר, רואים את ההרים מאחורי העץ הגדול מצד שמאל. בתמונה השמאלית, נבחר אחד לכל התמונה. פשוט נעלמו ההרים. הפונקציה לא ידעה להתמודד עם 3 גוונים ממש שונים: צבע ההרים, השמיים והצמחיה. בפונקציה האדפטיבית, זה נעשה פר פיקסל שמוקף בתת תמונה בגודל `WINSIZE`. לכן היה יותר קל שהגיעו חתיכות של תמונות.

## ד. השוואה בין `binarizeOptAdaptive()` ל `binarizeOptAdaptiveIH()`

○ תמונה מקורית מה `lakeScene.tif` : IMAGES



○ כצפוי אותה תוצאה. אבל מה מבחינת יעילות?

- אז מסתבר שחישוב ההיסטוגרמה בכל איטרציה לקח המון זמן מבחינת סך כל ריצת התוכנית. לשם מדידה השתמשנו ב TIC TOC וגם בפרופיילר

```
Command Window
>> ourLastTask7
time for binarizeOptAdaptive(img3,31) took:
Elapsed time is 722.042887 seconds.
time for binarizeOptAdaptiveIH(img3,31,IH) took:
Elapsed time is 362.504548 seconds.
the advantage is clearly when winsize is big. while in binarizeOptAdaptive we calc each subimg histogram
,in binarizeOptAdaptiveIH we do it once preprocessing. if win size if big, the work for binarizeOptAdaptive
grows significantly, while binarizeOptAdaptiveIH stays the same regarding the hist calc
>>
```

- זו התוצאה של הרצת הסקריפט (ד7) ורואים שגם בחלון קטן בגודל 31 הזמן משתפר.

- בלי IH: 722 שניות
- עם IH: 362 שניות
- ההפרש: 360 שניות.

- כל ההבדל בקוד הוא קריאה לפונקציה optimalQuantization עם ולי ההיסטוגרמה של התמונה שמועברת.

**Profile Summary**  
Generated 30-Nov-2016 22:03:03 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">ourLastTask7</a>	1	1099.717 s	0.070 s	
<a href="#">binarizeOptAdaptive</a>	1	722.046 s	4.915 s	
<a href="#">optimalQuantization</a>	65536	708.429 s	63.540 s	
<a href="#">histImage</a>	65536	365.535 s	365.535 s	
<a href="#">binarizeOptAdaptiveIH</a>	1	362.501 s	7.502 s	
<a href="#">optimalQuantizationUsingHistFromIH</a>	65536	337.036 s	62.286 s	
<a href="#">optimalQuantization&gt;calcQiByZi</a>	537927	148.406 s	148.406 s	
<a href="#">...tizationUsingHistFromIH&gt;calcQiByZi</a>	537927	147.314 s	147.314 s	
<a href="#">optimalQuantization&gt;calcE</a>	537927	93.884 s	93.884 s	
<a href="#">optimalQuantizationUsingHistFromIH&gt;calcE</a>	537927	91.841 s	91.841 s	
<a href="#">mean</a>	131072	17.354 s	17.354 s	

- תמונה של זמני הפונקציות.
- ניתן לראות שבלי IH לקח 722 שניות שמתוך 365 שניות בוזרזו על histImage.
- עם IH לקח 362 שניות. אם נסכום את הזמן שלקח ל histImage נקבל 727 שניות, כלומר כמעט את הזמן שלקח בלי IH (החמש שניות זה החישוב של ההיסטוגרמה מתוך IH + זמן העברת ההיסטוגרמה לכימות האופטימלי).