

שם הקובץ חייב להיות כשם הפונקציה - אחרת מטלב לא ימצא את הקובץ כשנקרא לפונקציה. אבל אפשר לרשום בתוך הקובץ עם שמות אחרים, והן יהיו פונקציות פנימיות - רק מתוך אותו קובץ אפשר לקרוא להן.

שיפור תמונה

אם ניקח את הקובץ darkimage.tiff נראה שהערכים שלו הם בין 20 ל-30 - והתמונה שחורה. אפשר למתוח לינארית את ההיסטוגרמה:

$$i = (1 - 20) * (255 / 10)$$

אבל זה עובד בגלל שההיסטוגרמה המקורית מתנהגת יפה (חוץ מזה שהיא צרה). זה לא בהכרח עובד בכל היסטוגרמה - למשל אם המידע מרוכז בקצוות.

Histogram Equalization

קודם כל צריך לבנות את ההיסטוגרמה:

```
[rows, cols] = size(i);  
hist = zeros(1, 256);  
for r = 1:rows  
    for c = 1:cols  
        hist(i(r, c) + 1) ++;  
    end  
end
```

היסטוגרמה צוברת

נרצה לקבל היסטוגרמה צוברת:

```
Ahist = hist;  
for color = 2:256  
    Ahist(color) = Ahist(color - 1) + hist(color);  
end
```

נירמול

כדי לא להיות תלויים בגודל התמונה, נרצה לנרמל את ההיסטוגרמה:

```
Ahist = Ahist / (rows * cols)
```

למה קודם צוברים ורק אחר כך מנרמלים?

כשמנרמלים עוברים לשברים עשרוניים ואז מאבדים דיוק.

למה בכלל צריך לצבור?

אנחנו רוצים ליצור וקטור המרה שיקבע לנו באיזה צבע להחליף כל צבע. כשההיסטוגרמה אחידה, ההיסטורגמה הצוברת היא לינארית (כי $\int a dx = ax + c$). מכיוון ההיסטוגרמה הצוברת היא מונוטונית לא-יורדת (גם המקור וגם היעד), זה עוזר לנו לבנות אלגוריתם יעיל לבנות את פונקציית ההמרה ביניהם.

איך בונים את וקטור ההמרה?

היסטוגרמת היעד היא:

```
Dhist = (1:255) / 255
```

עוברים על אינדקסים בהיסטוגרמת המקור והיסטוגרמת היעד, ומתאימים אינדקסים:

```
cv = zeros(256);  
s = 1;  
d = 1;  
while s < 257  
    if Ahist(s) <= Dhist(d)  
        cv(s) = d - 1;  
        s ++;  
    else  
        d ++;  
    end  
end  
end
```