אחסון יעיל של תמונות במאגר נתונים בעזרת טכניקות דחיסה

חבור על מחקר

לשם מלוי חלקי של הדרישות לקבלת התואר

מגיסטר במדעים

בהנדסת חשמל

מאת

אריאל לכטמן

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

חשון תשמ"ה **חיפה** נובמבר 1984

המחקר נעשה בהנחייתו של פרופסור דוד מלאך בפקולטה להנדסת חשמל.

תודתי העמוקה נתונה לפרופסור דוד מלאך על הנחייתו, תרומתו ומעורבותו המלאה והמסורה בכל שלבי המחקר.

תודתי נתונה לפרופסור יעקב זיו על תרומתו הרבה בהנחיית המחקר.

ברצוני להודות לציפי פורטנוי וליורם אור-חן על עזרתם הנדיבה במהלך עבודתי במעבדה לעיבוד אותות.

להורי ולעילית

תוכן הענינים

תוכן העניינים (המשך)

תקציר

בעבודה זו מוצעים ונבחנים מימושים חדשים של אלגוריתם זיו-למפל המיועדים לקידוד תמונות.

באלגוריתם זיו-למפל לקידוד אוניברסלי של סדרות לא נדרשת ידיעה מוקדמת של התכונות הסטטיסטיות של המקור ולא נגרם עוות למידע (Lossless Coding). כמו כן עבור מקורות סטציונריים ניתן להשיג בעזרתו קצב הקרוב כרצוננו לאנטרופית המקור. האלגוריתם הופעל על מקורות מידע שונים אך לא הוצגו עד כה תוצאות ישומיו על קידוד תמונות.

מטרת העבודה היא לפתח מימושים חדשים של האלגוריתם לקידוד יעיל של תמונות, דהיינו דחיסה מכסימלית וסיבוכיות מינימלית.

שלוש הגירסאות לממוש האלגוריתם המוצעות בעבודה זו מבוססות על עצי פיסוק ועל טבלת תרגום (המייצגת את עצי הפסוק) בתהליך הפיענוח. בעזרת המימושים מתקבל זמן קבוע לפענוח מילות הקוד, תכונה המאפשרת שליפה מהירה של התמונה. בגירסה הראשונה הקידוד מתבצע בנפרד על כל אחד ממישורי הסיביות של התמונות המיוצגות לפי קוד GRAY, ביצוג זה קיימת קורלציה גבוהה בין סיביות עוקבות במישורי הסיביות המשמעותיות ביותר, ועל ידי כך מתקבל יחס דחיסה טוב יותר מזה המתקבל על ידי המימושים הידועים של האלגוריתם. כמו כן ניתן לשחזר את התמונות באופן הדרגתי ועל ידי כך להאיץ את קצב חיפושן במאגרי נתונים.

בגירסה השניה משתמשים בעץ פיסוק קבוע, הנקרא עץ אוניברסלי, שבאמצעותו מתבצעים תהליכי הקידוד והפענוח. העץ האוניברסלי נבנה מראש בעזרת קבוצת תמונות אופייניות. בגירסה זו מתקבלות מילות קוד באורך קבוע וחסינותה בפני רעש גבוהה יותר מאשר במימושים מקובלים אחרים. כאשר פיסוק התמונות בעזרת שיטה זו נעשה על פי האלפבית המקורי של ערכי הפיקסלים, נמצא כי יחס הדחיסה רגיש לשנויים בתכונות הסטטיסטיות של התמונות. מכיוון שהתמונות אינן סטציונריות בדרך כלל, ביצועי האלגוריתם נפגעים.

הגירסה השלישית מהווה שילוב של שתי הגירסאות הקודמות, כלומר, הקידוד מתבצע על כל מישור סיבית בנפרד בעזרת עצים אוניברסליים. בנוסף ליתרונות של המימושים על פי שתי הגירסאות לעיל, רגישות מימוש זה לתכונות הסטטיסטיות של התמונה נמוכה.

במסגרת המחקר נבחן מודל ליצירת תמונה המבוסס על מקור מרקובי מסדר ראשון, שבעזרתו אנו מנסים להעריך ולהסביר את ביצועיהם של המימושים השונים שהוצעו. התקבלה אמנם התאמה טובה עבור הגירסה הראשונה אך להסברת תופעת הרגישות של העץ האוניברסלי עפ"י הגירסה השניה ואי הרגישות היחסית של הגירסה השלישית מצאנו כי יש צורך במודל מורכב יותר.

בניסויים שנערכו התקבל כי עבור תמונות פנים המיוצגות על ידי 8 סיביות לפיקסל ניתן להשיג יחס דחיסה מירבי של 2:1. כדי לשפר יחס זה הופעלו המימושים השונים של האלגוריתם על תמונות פנים המיוצגות על ידי 4 סיביות. ייצוג זה עדיין משמר את איכות התמונה ויחס הדחיסה שמתקבל מגיע עד כדי 3:1. את איכות התמונות המתקבלות ניתן עוד לשפר על ידי הוספת Dither.

על מנת להמשיך לשפר את תוצאות הדחיסה נבדק גם שילוב האלגוריתם עם טכניקות קידוד אחרות. מבין הטכניקות שנבדקו ראויה לציון התמרת הדמרד (Hadamard) שהופעלה תחילה על תמונות המיוצגות על ידי 8 סיביות לפיקסל. לאחר איפוס מקדמי ההתמרה בעלי השונות הנמוכה, המקדמים הנותרים עוברים כימוי (קוונטיזציה) ומקודדים על ידי האלגוריתם. בשיטה זו מתקבלת דחיסה של עד כדי סיבית אחת לפיקסל, כלומר יחס דחיסה 8:1, עם איכות תמונות משוחזרות טובה מאוד.

<u>רשימת סימונים וקיצורים</u>

X_1^N, x_1^N - סדרות משתנים באורך כללי N.

$\hat{x}_1^n$ - סדרת משתנים בינריים באורך n.

$\alpha = |A|$ - עוצמת אלפבית A כלשהוא.

λ - מחרוזת בעלת אורך אפס.

$X_{n_{j-1}+1}^{n_j}$ - מחרוזת j-ית המתקבלת מתהליך הפיסוק של סדרה כלשהיא.

ℓ_j - אורך (מס' הסימבולים) המחרוזת ה-j-ית.

p - מס' המחרוזות המתקבל מתהליך הפיסוק של סדרה כלשהיא.

p_B, p_b - ערכי P המתקבלים בעזרת עץ פיסוק בינרי רגיל, ועל ידי עץ פיסוק בינרי תוך התחשבות בגבולות ה-Byte, בהתאמה.

L_j - מס' הסיביות הדרוש לקידוד מילת הקוד ה-j-ית.

L - סה"כ מס' הסיביות המוקצב למילות הקוד המתקבלות מקידוד סדרה כלשהיא.

L^b, L^B, L^T - ערכי L המתקבלים על ידי עץ פיסוק מדרגת α , עץ פיסוק בינרי תוך התחשבות בגבולות ה-Byte ועץ פיסוק בינרי רגיל, בהתאמה.

$\rho_E(X_1^N)$ - יחס הדחיסה המתקבל מהפעלת המקדד E על הסדרה X_1^N.

$\rho(u)$ - מס' סיביות לפיקסל המתקבל לאחר קידוד לפי אלגוריתם זיו-למפל.

$\rho_{name}(u)$ - ערך $\rho(u)$ המתקבל על ידי המימוש (name) של אלגוריתם ZL.

k - מס' הסיביות הדרוש ליצוג סימבול כלשהוא של האלפבית.

EC_j - מס' הסיביות הנוספות הנלוות למילת הקוד ה-j-ית.

$H(u)$ - אנטרופיה מסדר אפס.

$H(\underline{u})$ - אנטרופיה של בלוק סימבולים.

$\hat{H}(u)$ - משערך האנטרופיה לפי אלגוריתם זיו-למפל.

$H(y_n/y_{n-1})$ - אנטרופיה מותנית.

u, X, Y - מקורות מידע.

ℓ - מישור סיבית.

רשימת סימונים וקיצורים (המשך)

a_i	- הסיבית ה-i-ית של משתנה בינרי כלשהוא.
Δ	- צעד הכימוי?
P_i	- הסתברות הופעת הסימבול $\pm i\Delta$ במודל ליצירת תמונה.
$B_\ell(i)$	- מ"א בינרי עבור מישור סיבית ℓ ומצב i של שרשרת מרקוב.
$P_i^\ell(u)$	- הסתברות לפליטת הסימבול u עבור מישור סיבית ℓ ומצב i של שרשרת מרקוב.
$H(Y_\ell)$	- אנטרופיה ממוצעת למישור הסיבית ℓ של המקור Y .
$f(i,j)$	- ערך הפיקסל (i,j) של התמונה המקורית.
$y(i,j)$	- ערך הפיקסל של התמונה לאחר עיבוד.
C_i	- המקדם ה-i-י של התמרת הדמרד (Hadamard).
מ"א	- משתנה אקראי - משתנים אקראים.
ZL	- קיצור לאלגוריתם זיו-למפל.
log x	- מייצג $\log_2 x$ אם לא צויין אחרת.
mod	- פעולת מודולו.

פרק 1 : מבוא

השימוש במידע שנמצא בתמונות המיוצגות באופן ספרתי הולך וגדל בתחומי המדע, הצבא והמסחר עקב התפתחות המחשב ורשתות התקשורת הסיפרתית.
הבעיה המרכזית בשימוש בתמונות אלה נובעת מכמות הזכרון הגדולה הנחוצה לצורך איחסונן במאגרי נתונים. כדי לצמצם את כמות הזכרון נעזרים בטכניקות דחיסה המופעלות על סמך עקרונות שונים.
מחקר זה עוסק בשיטות קידוד לדחיסת תמונות המבוססות על אלגוריתם זיו-למפל לקידוד אוניברסלי של סדרות נתונות [1].

1.1 תמונה ביצוג ספרתי

יצוג תמונות באופן ספרתי נעשה על ידי אלמנטי תמונה הנקראים פיקסלים (picture element = pixel).
ניתן לאפיין תמונה בשתי דרכים. האחת כמקור אינפורמציה בעל אלפבית U של 2^k אותיות, כאשר k הינו מס' הסיביות ליצוג הפיקסל, בדרך כלל $k = 8$ ואז מס' האותיות הוא 256. האות "0" מסמנת צבע שחור והאות "255" צבע לבן. תאור מפורט יותר של התמונה כמקור אינפורמציה נמצא בנספח א'.
האפשרות השניה לאפיון תמונה היא על ידי סדרת מספרים $x_1^N = x_1,\ldots,x_N$, $x_i \in U$ לכל i, כאשר N הינו מספר הפיקסלים בתמונה אחת או במספר תמונות. סריקת התמונה הדו-מימדית לקבלת סדרה זו נעשית על ידי מפוי ידוע מראש, שהמקובל ביותר הינו ה-raster scan.
בתהליך קידוד התמונה בעזרת אלגוריתם זיו-למפל (ZL) נתיחס לתמונה כאל סדרת מספרים x_1^N.
כדי להעריך את ביצועי טכניקות הקידוד השונות נשתמש בקבוצה של 3 תמונות פנים, אשר תצלומיהן מופיעים בנספח ד'.
הבחירה של קבוצת תמונות מסויימת (תמונות פנים) מאפשרת פיתוח מודל מתאים שבעזרתו נעריך את ביצועי אלגוריתמי הקידוד השונים. התיחסות לתמונות ככלל אינה נוחה, מפני שקיימת שונות גדולה בתכונות הסטטיסטיות של הקבוצות השונות.

1.2 קידוד תמונות

ניתן לחלק את שיטות הקידוד לשתי קבוצות עיקריות, קידוד ללא עוות וקידוד תוך גרימת עוות.

רוב שיטות הקידוד המופעלות על תמונות שייכות לקבוצה שניה שאף היא מתחלקת ל-3 תתי קבוצות, קידוד המבוסס על שיטות חיזוי, קידוד המבוסס על שיטות טרנספורמציה וקידוד המבוסס על שילובן של שתי גרסאות אלה, הנקרא Hybrid Coding [2].

הכנסת עוותים לתמונה המקודדת (או איבוד חלק מהמידע) מאפשרת השגת יחסי דחיסה טובים בשיטות אלה. תאור מקוצר וניתוח טכניקות קידוד רבות הגורמות לעוות נמצא במאמרי סקירה עדכניים [3], [4] ובספרות המקצועית השוטפת [5], [6], [7].

1.3 קידוד ללא עוות

בשיטות קידוד ללא עוות לא מוכנסת שגיאה בתהליך הקידוד. ניתן לסווגן לשתי קבוצות: א) קידוד כאשר התכונות הסטטיסטיות של המקור ידועות או כאשר בונים מודל לאפיונן. בשיטה זו מנסים למצוא את הקידוד האופטימלי (בממוצע) עבור קבוצות מסויימות של סימבולים, כאשר כל קבוצה מאופיינת על ידי ההסתברות להופעתה. האלגוריתם של Huffman [8] מייצג קבוצה זו.

אלגוריתם הקידוד של Huffman איננו יעיל לקידוד תמונות מפני שהשימוש בו מחייב ידיעה מוקדמת של התכונות הסטטיסטיות של התמונות (הסתברות הופעת הסימבולים). דרישה זו איננה מתאימה לאי הסטציונריות המאפיינת את התמונות. שיטות קידוד נוספות המבוססות על אותו עיקרון הן B Codes ו-Shift Codes [5], אך הן אינן אופטימליות מבחינת יכולת הדחיסה. שיטות הקידוד לפי "Run Length" [5] שייכת לקבוצה זו אך יעילותה מוגבלת לתמונות בינריות.

ב) קידוד סדרות ללא צורך בידיעה מוקדמת של התכונות הסטטיסטיות שלהן או בהנחה כלשהיא לגביהן (Coding of individual sequences). אלה הם אלגוריתמים בהם תהליך האופטימיזציה של הקידוד מתבצע אך ורק על סמך סדרות סימבולים המוכנסות למקדד.

בעזרת אלגוריתם זיו-למפל לקידוד אוניברסלי של סדרות השייך לקבוצה זו ניתן לדחוס מידע ללא עוות וללא צורך בידיעה מוקדמת של התכונות הסטטיסטיות של המקור. תכונות אלה מצביעות על כך שלאלגוריתם יכולת מעשית לקידוד יעיל של תמונות.

מטרת המחקר

מטרת המחקר היא לפתח ולבחון מימושים חדשים של האלגוריתם לצורך השגת קידוד יעיל של התמונות, דהיינו דחיסה מכסימלית וסיבוכיות מינימלית, על ידי התאמת המימושים לתכונות הסטטיסטיות של התמונות.

1.4 מדידת אנטרופיה

החסם התחתון של מספר הסיביות לאות מקור שניתן להשיג על ידי קידוד ללא עוות, נתון על ידי האנטרופיה של המקור שהיא גודל חשוב, אליו משווים את מספר הסיביות לאות מקור המושג באמצעות אלגוריתמים שונים לקידוד של מקורות אינפורמציה ללא עוות. קיימות מספר נוסחאות לחישוב האנטרופיה התלויות בתכונות הסטטיסטיות של המקור. תאור נוסחאות אלה נתון בנספח ב'. בנספח מוגדר גם גודל נוסף $\hat{H}(u)$, שהוא משערך האנטרופיה המתקבל בעזרת אלגוריתם זיו-למפל. $\hat{H}(u)$ שואף לאנטרופיה האמיתית של המקור [9] ומהווה חסם תחתון לתוצאות הדחיסה שניתן להשיג על ידי אלגוריתם זיו-למפל ללא תלות במימוש.

1.5 <u>הפעלת אלגוריתם זיו-למפל על תמונות פנים</u>

כדי לקבוע את יכולת הדחיסה של האלגוריתם על תמונות פנים המיוצגות על ידי 8 סיביות לפיקסל נמדדו ערכי האנטרופיה הבאים: $H(u)$ - אנטרופיה מסדר אפס;
$H(\underline{u})$ אנטרופיה של בלוק סימבולים, כאשר המימד של וקטור $\underline{u}$ הינו 2;
$H(u_i/u_j)$ - אנטרופיה מותנית, ו-$\hat{H}(u)$ המשערך לאנטרופיה.
מתוצאות המדידה המוצגות בנספח ב' מתקבל:

$$\hat{H}(u) \approx H(u_i/u_j) < H(\underline{u}) < H(u)$$

המעיד על התלות החזקה שקיימת בין פיקסלים סמוכים ועל כך שהדחיסה על ידי אלגוריתם ZL טובה מזו שניתנת להשגה על ידי אלגוריתם Huffman. כמו כן התקבל שממוצע הערך $\hat{H}(u)$ עבור 3 תמונות פנים הינו כ-4.2 סיביות לפיקסל. דחיסה זו אינה מספקת בדרך כלל.

<u>תמונות המיוצגות על ידי 4 סיביות</u>

אחת הדרכים להגדלת יחס הדחיסה היא לייצג מראש את התמונה ב-4 הסיביות המשמעותיות ביותר בלבד. למרות העוות הנובע מהכימוי (קוונטיזציה) ל-4 סיביות, איכות תמונות הפנים המיוצגות על ידי 4 סיביות בלבד הינה טובה מאוד. נוצרים מעט false countoring אשר ניתנים לטשטוש על ידי הוספת Ordered dither [10].
קיצוץ של 4 הסיביות הפחות משמעותיות מחייב שימוש בכל התחום הדינמי של מערכת הדגימה, או הפעלת עבוד מתאים להשוואת ההסטוגרמה לאחר הדגימה [6,פרק 12], זאת כדי שרוב האינפורמציה תמצא בסיביות היותר משמעותיות.
נמצא גם שהערך הממוצע של $\hat{H}(u)$ עבור תמונות אלה הוא כ-1.4 סיביות לפיקסל. ערך זה הינו החסם התחתון אליו נשאף להתקרב על ידי המימושים השונים של האלגוריתם תוך שמירה על איכות תמונה בעלת 16 רמות אפור שהיא איכות טובה בהחלט עבור תמונות פנים.
בגוף העבודה נציג את המימושים השונים של האלגוריתם, את ניתוח סיבוכיותם ואת ערכי הדחיסה הניתנים להשגה בעזרתם. נשווה את ערכי הדחיסה ל-$\hat{H}(u)$. כמו כן נבדוק את שילוב האלגוריתם עם טכניקות קידוד אחרות, על מנת לקבל יחס דחיסה טוב יותר.

1.6 מבנה העבודה

פרק 2 - בפרק זה מתואר אלגוריתם זיו-למפל ומוצגים מימושיו היסודיים המבוססים על שימוש בעצי פיסוק. התאור מלווה בדוגמאות, בניתוחים אנליטיים של סיבוכיות המימושים ובהצגת ערכי הדחיסה המתקבלים בעזרתם. פרק זה מהווה בסיס לתאור המימושים החדשים.

פרק 3 - פרק זה מכיל את התאור של 3 המימושים החדשים של האלגוריתם את הניתוחים האנליטיים של סיבוכיותם ואת ערכי הדחיסה המתקבלים בעזרתם. בחלק האחרון של הפרק נבדקת רגישות תוצאות הדחיסה של אחד המימושים לתכונות הסטטיסטיות של התמונה.

פרק 4 - בפרק זה מוצע ומנותח מודל ליצירת תמונה בעזרתו אנו מעריכים ומסבירים את ביצועיהם של המימושים החדשים.

פרק 5 - בפרק זה נבדקת יכולת הדחיסה של אלגוריתם ZL בשילוב עם טכניקות קידוד מקובלות הגורמות לעוותים בתמונה.

פרק 6 - סכום ומסקנות.

פרק 2 : אלגוריתם זיו-למפל

אלגוריתם זיו-למפל לקידוד אוניברסלי של סדרות נתונות [1] הינו אלגוריתם לדחיסת מידע ללא הכנסת עוותים (Lossless Coding), כאשר תהליך האופטימיזציה של הקידוד (למידת תכונות הסדרה) מתבצע אך ורק על סמך סדרת הסימבולים שמוכנסת למקדד. האלגוריתם מתבסס על עיקרון פיסוק אינקרמנטלי אשר מתבצע באופן רציף על סדרת הכניסה, ועל קידוד מתאים (מילות קוד באורך משתנה) למחרוזות שנוצרות בתהליך הפיסוק. בפרק זה נציג את האלגוריתם המקורי ונתאר וננתח מספר מימושים בסיסיים המסתמכים על עצי פיסוק.

2.1 תאור האלגוריתם

המקדד E מוגדר על ידי החמישיה (S,A,B,g,f) כאשר S: אוסף סופי של מצבים, A: א"ב (סופי) של אותיות הכניסה בעל עוצמה α (מס' האותיות בא"ב) ($|A| = \alpha$), B: אוסף סופי של מילות קוד מעל א"ב סופי (בדרך כלל הא"ב בינרי), g: פונקצית "המצב הבא", $g: S\times A \to S$. הפונקציה קובעת את המצב הבא על סמך המצב הנוכחי של המקדד ועל סמך קלט הכניסה ל-E. f: פונקציה המייצרת מילות קוד, $f: S\times A \to B$.

המקדד E משתמש באוטומטה מהסוג ILF (Information lossless of Finite Order) [11], והוא מחולק לשני חלקים. חלק ראשון, המקודד "החיצוני" שמקבל בכניסתו בלוקים של אותות מקור בעלי אורך קבוע N, $N >> 1$. נסמן ב-x_1^N את הבלוק הראשון של סדרה אינסופית x מעל A. המקדד פולט מילות קוד בעלות אורך משתנה מעל B. מילות קוד אלה מופקות על ידי המקדד "הפנימי" (inner code). המקדד החיצוני מבצע (reset) איפוס וחוזר למצבו ההתחלתי לאחר טיפול בכל בלוק בעל אורך N. תהליך הקידוד מתחיל מחדש עבור כל בלוק. החלק השני של המקדד E הוא המקדד "הפנימי", שמקבל בכניסתו מחרוזות בעלות אורך משתנה, שבדרך כלל אורכן הולך וגדל. הקידוד של מחרוזות אלה נעשה באופן עוקב (sequential) והוא תלוי במצב המקדד עם קבלת כל מחרוזת. מחרוזות אלה נגזרות מתוך סדרת הכניסה למקדד החיצוני על ידי תהליך של פיסוק אינקרמנטלי. תהליך זה סורק את סדרת אותיות הכניסה ומייצר מחרוזת חדשה על ידי סימון "פסיק" (,) כאשר מתגלה אוסף של אותיות שטרם הופיעו. אוסף אותיות אלה נקרא בתהליך זה קידומת.

2.1.1 תיאור מפורט וניתוח ביצועי האלגוריתם

תהליך הפיסוק מסומן על ידי:

$$X_1^N = X_{n_o+1}^{n_1},\quad X_{n_1+1}^{n_2},\ \ldots,\ X_{n_p+1}^{n_{p+1}} \qquad n_o \triangleq 0, \quad n_{p+1} \triangleq N$$

והפיסוק נקרא אינקרמנטלי אם p המחרוזות הראשונות $X_{n_{j-1}+1}^{n_j}$, $1 \le j \le p$

כולן שונות, ובנוסף קיים מספר שלם i, $j > i$ עבור כל j, $j = 1,2,\ldots,p+1$,

כאשר $n_j - n_{j-1} > 1$ כך ש-

$$X_{n_{i-1}+1}^{n_i} = X_{n_{j-1}+1}^{n_j-1}.$$

באתחול הפיסוק מניחים שקיימת המחרוזת בעלת אורך אפס λ, $\lambda \triangleq X_{n_{-1}}^{n_o}$;

דהיינו ניתן לרשום $X_1^N = \lambda X_1^N$.

כדי לקבוע את מחרוזת ה-j-ית, $1 \le j \le p+1$ נבחר את n_j כמספר השלם הגדול

ביותר, לא גדול מ-N, אשר מקיים:

$$X_{n_{i-1}+1}^{n_i} = X_{n_{j-1}+1}^{n_j-1}.$$

המחרוזת ה-j-ית מורכבת מ-$k = n_j - (n_{j-1}+1)$ סימבולים, כאשר $k-1$ הסימבולים

הראשונים כבר הופיעו במחרוזת ה-i-ית. הסימבול ה-n_j הינו הסימבול הנוסף

אשר גורם לכך שהמחרוזת ה-j-ית שונה מכל המחרוזות האחרות.

קידוד של $X_{n_{j-1}+1}^{n_j}$ הינו יצוג בינרי של המספר I, כאשר:

$$I(X_{n_{j-1}+1}^{n_j}) \triangleq i\cdot\alpha + I_A(X_{n_j}), \qquad \alpha = |A| \tag{2.1}$$

I_A הינו מיפוי של א"ב הכניסה A אל תוך קבוצת המספרים השלמים 0 עד $\alpha-1$.

משום ש-$0 \le i \le j-1$ נקבל ש-

$$0 \le I(X_{n_{j-1}+1}^{n_j}) \le (j-1)\alpha + \alpha - 1 = j\alpha - 1$$

ומספר הסיביות הדרושות לקידוד המילה ה-j-ית הינו: $L_j = \lceil \log\,(j\alpha) \rceil$,

$\log x$ מייצג $\log_2 x$ אם לא צויין אחרת,

וכאשר $\lceil x \rceil$ מציין את הערך השלם הגדול ביותר שכולל את x.

בהנחה שבבלוק x_1^N יש $p+1$ מחרוזות שונות נקבל שהמספר הכולל של הסיביות הדרושות לקידוד x_1^N הינו:

$$L = \sum_{j=1}^{p+1} L_j = \sum_{j=1}^{p+1} \lceil \log(j\alpha) \rceil ,$$

על סמך אי-השוויון :

$$\lceil \log x \rceil \le 1 + \log x = \log 2x$$

והביטוי הנ"ל נרשום את השוויונות הבאים:

$$L \le \sum_{j=1}^{p+1} \log(2\alpha j) < (p+1) \log(2\alpha(p+1)) .$$

נקבל שיחס הדחיסה שניתן להשיג בעזרת המקדד E עבור x_1^N נתון על ידי:

$$\rho_E(x_1^N) \le \frac{p+1}{N \log\alpha} \log(2\alpha(p+1)) \tag{2.2}$$

כמו כן במאמר [1] מופיעות התוצאות הבאות:

א) דחיסה של סדרה אינסופית x מעל A , בעזרת המקדד E בבלוקים בעלי אורך N כל אחד נתונה על ידי הביטוי הבא:

$$\rho(x) = \limsup_{N \to \infty} \limsup_{k \to \infty} \frac{1}{kN\log\alpha} \sum_{i=0}^{k} p(x_{iN+1}^{(i+1)N}) \log p(x_{iN+1}^{(i+1)N}) \tag{2.3}$$

ב) הוכח שאם הסדרה x נלקחה ממקור ארגודי בעל אנטרופיה H אזי:

$$\Pr\{\rho(x) = H\} = 1$$

ג) עבור מקור סטציונרי בעל אנטרופיה H נקבל: $E[\rho(x)] = H$, E מסמן תוחלת.

<u>תמצית האלגוריתם:</u>

להלן תמצית האלגוריתם לקידוד סדרה אינסופית x מעל A :

0 - איתחול. הכנס את המחרוזת בעלת אורך אפס λ לקבוצת המחרוזות ההתחלתית ("המילון"). קבע $j = 1$.

1 - החל מהמקום הנוכחי של המחוון בסדרת הכניסה, מצא את המחרוזת הארוכה ביותר אשר שייכת למילון:

$$.(X_{n_{j-1}+1}^{n_j-1} = X_{n_{i-1}+1}^{n_i})$$

2 - פלוט את הקידוד של המספר המזהה i על המחרוזת במילון המתאימה למחרוזת שנמצאה בסדרת הכניסה יחד עם הקידוד של הסימבול הראשון המופיע לאחר המחרוזת שבסדרת הכניסה.

3 - הוסף מחרוזת חדשה למילון המורכבת מהמחרוזת שנמצאה בשלב (1) ומהסימבול שאחריה. קבע $j = j+1$. הקצב למחרוזת החדשה ערך מזהה j.

4 - קדם את המחוון של סדרת הכניסה מעבר למחרוזת החדשה אשר כוללת את הסימבול הנוסף. חזור לשלב (1) אם המחוון לא מצביע על סוף הבלוק.

5 - בסוף הבלוק חזור לשלב (0).

<u>דוגמא:</u> נתונה סדרה x מעל A, $A = \{0,1\}$, $\alpha = |A| = 2$. נתבונן ב-11 האותיות הראשונות של הסדרה: $X = 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0$.

נקבע את המיפוי I_A:

$$I_A(x_{n_j}) = \begin{cases} 0 & x_{n_j} = 0 \\ 1 & x_{n_j} = 1 \end{cases}$$

לאחר תהליך הקידוד נקבל:

1. פיסוק הסדרה:		λ,	0,	01,	011,	1,	11,	00,
2. ערך j של המחרוזת במילון.	j	1	2	3	4	5	6	7
3. ערך מזהה של המחרוזת הקודמת במילון.	i		1	2	3	1	5	2
4. מילות הקוד.	$I(X_{n_{j-1}+1}^{n_j})$		2	5	7	3	11	4

2.1.2 סיבוכיות האלגוריתם

שלב (1) של האלגוריתם הנ"ל קובע את יעילותו, בכך שזמן החיפוש קובע את קצב יצירת מילות הקוד. זמן החיפוש נמצא ביחס ישר לאורך המחרוזת שעוברת קידוד ולאורך סדרת הכניסה שכבר עברה קידוד.

שלב (3) קובע את כמות הזכרון הדרושה, שהיא ביחס ישר לאורך סדרת הכניסה שכבר עברה קידוד.

בהמשך העבודה נתיחס לשני הפרמטרים הבאים בניתוח הסיבוכיות של הישומים השונים של האלגוריתם: 1) זמן יצירת מילת קוד, 2) כמות הזכרון הדרושה.

2.1.3 המפענח

במאמר של זיו-למפל [1] לא נמצא ניתוח מפורט של ביצוע או תכונות המפענח. המחברים מסתפקים בהצגת המפענח אשר מוכיחה שקיים מיפוי הפוך למקדד.

2.2 תאור המימושים השונים של אלגוריתם זיו-למפל

מטרות עקריות במחקר הישומים השונים לאלגוריתם:

1) חיפוש יעיל של מחרוזת חדשה בתהליך הפיסוק, הווה אומר השגת קצב מכסימלי של יצירת מילות קוד.

2) הקצבה מינימלית של סיביות למילות קוד.

3) כמות זכרון מינימלית עבור תהליך הפיסוק.

4) מפענח בעל סיבוכיות נמוכה המאפשר פיענוח מהיר וכמות זכרון מינימלית.

2.2.1 מימוש מס' 1: מימוש האלגוריתם על ידי עץ פיסוק

עץ פיסוק בעל α ענפים בכל צומת דומה במבנהו לעץ חיפוש מדרגת α [12]. העץ בנוי כך שכל צומת מתאים לקידומת מסויימת. העץ מכיל בשורשו את הסימבול λ. מספר הענפים בכל צומת תלוי בעוצמת α של אלפבית A, $|A| = \alpha$.

תפקודו של עץ פיסוק לשמור אינפורמציה כך שיוכל לזהות את הקידומות השונות. נעזר במבנה של עץ פיסוק על מנת להקציב מילות קוד לכל מחרוזת. ההקצבה הינה כדלהלן:

1) לכל עלה נקבע מספר אשר ישמש כמילת קוד למחרוזת שנוצרה.

2) קביעת מספר לכל עלה נעשית על ידי העתקת מצב המונה אשר תוכנו גדל ב-1 עם יצירת כל ענף חדש בעץ. במקרה שעלה חדש מתקבל על ידי הרחבת צומת בעזרת אותו סימבול שדרכו מגיעים לצומת הנ"ל, המספר שמוקצב לעלה החדש זהה לזה של הצומת שממנו הוא יצא.

תאור האלגוריתם בצורתו הפורמלית נמצא בנספח ג' .

<u>דוגמא לביצוע הקידוד</u>

נתונה סדרה X מעל A,

נתבונן ב-8 האותיות הראשונות של הסדרה:

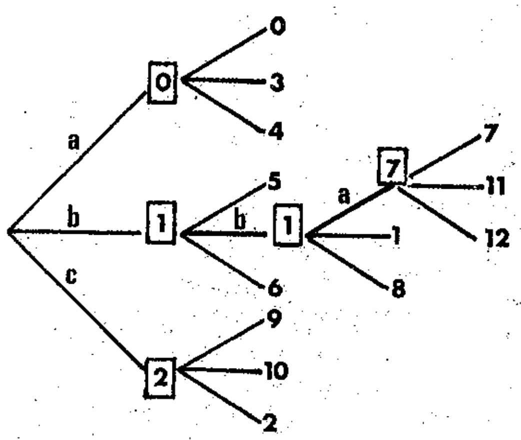
X = a b b b c b b a . . .

לאחר תהליך הקידוד נקבל:

1. פיסוק הסדרה: $= \lambda, a, b, bb, c, bba,$

2. מילות הקוד: 0 1 1 2 7

3. עץ הפיסוק שנוצר הינו:



$\boxed{i}$ מילת הקוד שנשלחה למפענה.

יש לשים לב שלמרות שלשתי מחרוזות שונות הוקצב אותו ערך מספרי, לא נוצרת בעיה בפיענוח משום שהמפענח יפרש אותן כשתי מילות קוד שונות. ראה בהמשך הסבר על תהליך הפענוח.

על פי האלגוריתם הנ"ל מתקבל שכאשר נוצרת מחרוזת חדשה מתווספים לעץ $(\alpha-1)$ עלים חדשים. מכך נובע ששיטת המספור של עלי עץ הפיסוק אקויוולנטית להקצבת מספר "רץ", בכפולות של $(\alpha-1)$ למחרוזות השונות.

בדוגמא הנ"ל: $\alpha - 1 = 2$.

פיסוק הסדרה	λ ,	a ,	b ,	bb ,	c ,	bba ,	. . .
מילת קוד: $(\alpha-1)j$	$2\cdot1$	$2\cdot2$	$2\cdot3$	$2\cdot4$	$2\cdot5$	$2\cdot6$	. . .

הקצבת סיביות

מספר הסיביות הדרוש לקידוד המחרוזת ה-j-ית הינו:

$$L_j^T = \lceil \log((\alpha-1)j) \rceil \tag{2.4}$$

ועבור קידוד של $p+1$ המחרוזות שנוצרו בתהליך הקידוד של הבלוק x_1^N דרושות:

$$L^T = \sum_{j=1}^{p+1} \lceil \log((\alpha-1)j) \rceil \quad \text{סיביות} \tag{2.5}$$

מהביטוי הנ"ל נובע שמספר הסיביות הדרוש לקידוד סדרה x_1^N בשיטה זו קטן יותר מאשר בגירסה המקורית מכיוון שתמיד מתקיים אי השוויון הבא:

$$\sum_{j=1}^{p+1} \lceil \log((\alpha-1)j) \rceil < \sum_{j=1}^{p+1} \lceil \log(\alpha j) \rceil \tag{2.6}$$

זמן יצירת מילת קוד:

בשיטת פיסוק זו מושג שיפור בזמן חיפוש מחרוזת בסדרת הכניסה ל-E. זמן זה נמצא ביחס ישר לאורך המחרוזת העוברת קידוד ולא לחלק של הסדרה שכבר עבר קידוד.

זמן עידכון העץ הנובע מהוספת α ענפים עם קבלת כל מחרוזת נתון על ידי $O(\alpha)$.

מכאן שזמן יצירת מילת הקוד ה-j-ית נתון על ידי $O(\ell_j + \alpha)$,

$\ell_j \triangleq n_j - n_{j-1}$ אורך (מספר סימבולים) של המחרוזת ה-j-ית.

<u>זכרון דרוש:</u>

כמות הזכרון הדרושה עולה ביחס ישר למספר המחרוזות שנוצרות בפיסוק של סדרת הכניסה x_1^N וביחס לעוצמת הא"ב α, דהיינו, כמות זכרון הדרושה לקידוד $M_R \triangleq$ נתונה על ידי הביטוי:

$$M_R \propto (p+1)\cdot\alpha \tag{2.7}$$

ביטוי זה מציג את אי-יעילותה של שיטה זו לקידוד סדרות מעל אלפבית בעל עוצמה גדולה. לדוגמא, בקידוד תמונות בעלות 8 $\frac{\text{סיביות}}{\text{פיקסל}}$, $\alpha = 256$, כמות הזכרון הדרושה לאחר פיסוק של 256 המחרוזות הראשונות שווה לכמות הזכרון הדרושה לאיחסון התמונה המקורית.

מגבלת גודל הזכרון של המחשב גורמת לכך שרק עבור $\alpha = 2$ הישום הזה יעיל. במקרה ש-$\alpha = 2$ מספר הסיביות הדרושות לקידוד סדרה x_1^N מעל $A = \{0,1\}$ נתון על ידי

$$L = \sum_{j=1}^{p+1} \lceil \log j \rceil,$$

כאשר $p+1$ מספר המחרוזות המתקבלות לאחר הפיסוק. כמות הזכרון הדרושה נתונה על ידי $M_R \propto 2(p+1)$.

<u>המפענח</u> (עבור $\alpha = 2$):

עקרונות הפיענוח מופיעים בהמשך. בנספח ד' נמצא פרוט של אלגוריתם הפיענוח. אלגוריתם הפענוח מבוסס על עץ פיענוח זהה לעץ הקידוד. עם קבלת מילת הקוד המפענח מחפש את המסלול (היחיד) שבין השורש לעלה בעל מילת הקוד שהתקבלה. פלט המפענח מורכב מהסימבולים (אפסים ואחדים) שבעזרתם מגיעים מהשורש לעלה מסויים. כמו כן, עם קבלת מילת הקוד מתווספים לעץ שני ענפים לצומת המתאים למילת הקוד הנ"ל, ועל ידי כך העץ מורחב.

ניתוח יעילות המפענח:

יעילות הפענוח נקבעת על פי זמן החיפוש של מילת הקוד בעץ. מבנה נתונים מתאים לחיפוש מהיר הינו "טבלת תרגום". מפתח הכניסה לטבלה הוא מילת הקוד. בכל כניסה של טבלת התרגום מופיעה סדרת סימבולים הנחוצה, בעת תהליך הפיסוק (פענוח), כדי להגיע מהשורש לעלה אשר מתאים למילת הקוד שהתקבלה. מתאור זה נובע שזמן החיפוש של מילת הקוד בטבלה נתון על ידי $O(1)$. כמות הזכרון הדרושה לאיחסון טבלת התרגום היא ביחס ישר למספר מילות הקוד, $M_R \propto 2(p+1)$. רוחב של כניסה בטבלת התרגום מוכתב על ידי אורך המחרוזת המתאימה לכניסה זו. לכן יש להגביל את האורך המכסימלי של המחרוזת כדי לאפשר מימוש מעשי של טבלה זו. בנספח ה' מתואר אלגוריתם לקידוד סדרה מעל א"ב בינרי עם הגבלת אורך המחרוזת.

2.2.2 מימוש מס' 2: קידוד סדרות מעל אלפבית עם $\alpha > 2$ בעזרת עץ פיסוק בינרי

על פי ניתוחי הסיבוכיות של מימוש מס' 1 ניתן לראות שעץ הפיסוק מעל א"ב בעל עוצמת $\alpha >> 2$ לא יעיל למימוש מעשי משום שכמות הזכרון הדרושה היא רבה מאוד $(M_R \propto (p+1)\alpha)$. לכן מתעוררת השאלה האם קיימת אפשרות לפסק את היצוג הבינרי של x_1^N מעל A שנסמנו ב-$\hat{x}_1^n$ בעזרת עץ פיסוק בינרי ויחד עם זאת לשמור על התכונות הסטטיסטיות של הסדרה המקורית,

x_1^N, כאשר $n = k \cdot N$ ו- $k = \lceil \log \alpha \rceil$.

נבדוק שתי אפשרויות:

א) פיסוק של $\hat{x}_1^n$ בעזרת עץ בינרי רגיל,

ב) פיסוק של $\hat{x}_1^n$ בעזרת עץ בינרי תוך שמירה על גבולות ה-"byte", כאשר לצורך פישוט הניסוח נגדיר $\text{byte} \triangleq$ אוסף k סיביות הדרוש ליצוג סימבול כלשהו של א"ב A ו- $k = \lceil \log \alpha \rceil$.

<u>דוגמא:</u> נתונה סדרה x_1^N מעל A,

$$k = \lceil \log 3 \rceil = 2,\ A = \{a,b,c\},$$

נקבע שהיצוג הבינרי של A הינו $a = 00$, $b = 11$, $c = 10$,

$$x_1^N = \ a \quad b \quad b \quad b \quad c \quad b \quad b \quad a \quad \ldots.$$

$$x_1^n = \ 00 \quad 11 \quad 11 \quad 11 \quad 10 \quad 11 \quad 11 \quad 00 \quad \ldots.$$

2.2.3 <u>שיטה א': קידוד בעזרת עץ פיסוק בינרי</u>

בשיטה זו מתיחסים ל-$\hat{x}_1^n$ כאל סדרה "רגילה" מעל א"ב בינרי ומבצעים את תהליך הקידוד של ישום מס' 1 כאשר $\alpha = 2$. בשיטה זו לא מנצלים את התכונות הסטטיסטיות של הסדרה המקורית x_1^N; הווה אומר, תכונות כמו סטציונריות וקורלציה בין איברים השונים של הסדרה לא מוצאות ביטוי ביצוג החדש של הסדרה במיוחד כאשר מדובר ב-α גדול. לאחר הפעלת ישום האלגוריתם בשיטה זו על הסדרה שבדוגמא הנ"ל נקבל:

$$\lambda\ ,0\ ,01,1\ ,1\ \ 1\ ,1\ \ 1\ \ 1\ ,0\ \ 1\ \ 1\ ,1\ \ 1\ \ 0\ ,0$$

j: 1 2 3 4 5 6 7 8

גבולות ה-"byte".

2.2.4 <u>שיטה ב': פיסוק בעזרת עץ בינרי תוך שמירה על גבולות ה-byte.</u>

עיקרון השיטה הזא לשמור על התכונות הסטטיסטיות של הסדרה המקורית x_1^N. ישום העקרון נעשה על ידי שמירת גבולות ה-"byte" בתהליך הפיסוק של היצוג הבינרי $(\hat{x}_1^n)$ של הסדרה x_1^N. מספר המחרוזות השונות שמתקבלות על ידי שיטה זו זהה לזה של ישום מס' 1 על הסדרה המקורית x_1^N , (פיסוק בעזרת עץ עם α ענפים בכל צומת). דימוי הפיסוק המקורי הושג על ידי כך שלאחר שפיסוק "טבעי" הוכנס לסדרה הבינרית $\hat{x}_1^n$ ממשיך תהליך גידול העץ וסריקת הסדרה x_1^n עד שהמחוון של תהליך הסריקה מגיע למקום שהוא כפולה של k (מספר הסיביות בכל byte).

<u>תמצית האלגוריתם:</u>

(0) התחל את עץ הפיסוק עם 2 ענפים כך שיכלול את הסימבולים '0' ו-'1'.
קבע $j = 2$.

(1) החל מהמקום הנוכחי של המחוון בסדרת הכניסה, מצא את המחרוזת הארוכה ביותר אשר שייכת לעץ הפיסוק.

(2) פלוט את הקידוד של המספר המזהה של המחרוזת שנמצאה, דהיינו את המספר המזהה של העלה.

(3) הוסף 2 עלים חדשים לעץ בצומת האחרון שאליו מגיעים בתהליך הפיסוק. הקצב לעלה שאליו ניתן להגיע דרך אותו סימבול כניסה של הצומת שלפניו את אותו הערך של הצומת. לעלה השני קבע ערך j. קבע $j = j + 1$.

(4) אם המחוון בסדרת הכניסה <u>מצביע</u> על סיום של byte, אז:

(A) קדם את המחוון מעבר למחרוזת החדשה.

(B) חזור לשלב (1) אם המחוון לא מצביע על סוף הבלוק.

(5) אם המחוון בסדרת הכניסה <u>לא מצביע</u> על סיום של byte, אזי:

(A) קדם את המחוון במקום אחד.

(B) פלוט את הסימבול הנוכחי (bit).

(C) התקדם, בעץ הפיסוק, לעלה החדש אשר מתאים לערך של הסימבול הנוכחי.

(D) חזור לשלב (3) אם המחוון לא מצביע על סוף הבלוק.

(6) בסוף הבלוק חזור לשלב (0).

<u>דוגמא:</u> נחזור לדוגמא הקודמת כדי להמחיש את פעולת האלגוריתם. על מנת להבהיר את הדוגמא נשתמש בסימונים הבאים:

[שלב (x)] : מצביע על השלב x, השלב בו מתבצעת הפעולה באלגוריתם.

: גבולות ה-"byte".

, : פיסוק תוך שמירת גבול ה-byte ([שלב (4A)] או [שלב (5A)]).

. : פיסוק "טבעי" לפי עץ בינרי רגיל [שלב 1].

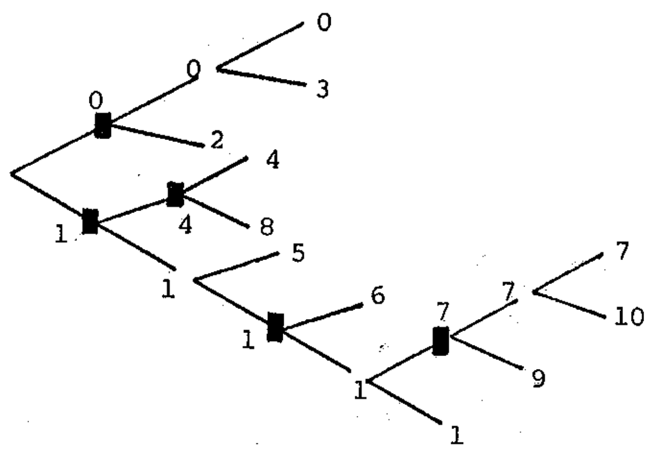
נקבל אז: $x_1^N = a , b , b \;\; b , c , b \;\; b \;\; a ,$

(A) פיסוק הסדרה: $x_1^n = \lambda,0.0,\ 1.1,\ 11\ \ 1.1,\ 10\ ,\ 11\ \ 11\ \ 0.0,$

(B) מילות הקוד וסיביות נוספות: $0\underline{0} \quad 1\underline{1} \quad 1\underline{1} \quad 4 \quad 7\underline{0}$ מילות קוד [שלב (2)]

סיביות נוספות [שלב (5B)]

(C) עץ הפיסוק שנוצר:



לכל עלה קיים מספר המזהה אותו. מילות הקוד הנפלטות מסומנות ב-(■) הן מספרי העלים שמתקבלים בעת קידוד הסדרה.

<u>הקצבת סיביות:</u>

מספר הסיביות הדרושות עבור מילת הקוד ה-j-ית (כולל הסיביות הנוספות שנשלחו) הוא:

$$L_j^B = \lceil \log (j + \sum_{i=1}^{j-1} EC_i) \rceil + EC_j \tag{2.8}$$

כאשר $EC_j \triangleq$ מספר הסיביות שנשלחו יחד עם מילת הקוד ה-j-ית; (סיביות שבין הפסיק "." לפסיק ","). כמו כן נגדיר: $\sum_{i=1}^{0} EC_i \triangleq 0$.

המספר הכולל של הסיביות הדרוש לקידוד הסדרה $\hat{x}_1^n$ נתון על ידי הביטוי:

$$L^B = \sum_{j=1}^{p+1} \{ \lceil \log (j + \sum_{i=1}^{j-1} EC_i) \rceil + EC_j \} \tag{2.9}$$

כאשר $p+1$ הוא מספר המחרוזות השונות שמתקבלות לאחר פיסוק של x_1^N לפי הגירסה המקורית של האלגוריתם.

<u>זכרון דרוש:</u>

כמות הזכרון הדרושה גדלה ביחס ישר למספר המחרוזות $(p+1)$ שנוצרות בפיסוק של הסדרה x_1^n וביחס ישר למספר הסיביות הנוספות EC_j שבין הפסיק הטבעי לפסיק המאולץ. הקשר הנ"ל ניתן לביטוי על ידי :

$$.M_R \propto 2((p+1) + \sum_{j=1}^{p+1} EC_j) \tag{2.10}$$

<u>זמן יצירת מילות קוד:</u>

זמן קידוד המחרוזת ה-j-ית נתון על ידי $O(\ell_j + EC_j)$ כאשר ℓ_j הוא האורך של המחרוזת ה-j-ית.

<u>מפענח</u>

במימוש זה סיבוכיות הפיענוח של מילות הקוד גדולה מזו שבמימוש מס' 1. (עץ פיסוק מדרגת α), מפני שכאן יש לטפל במילות הקוד j , וגם בסיביות הנוספות EC_j.

עקרונות תהליך הפיענוח מפורטים להלן:

עם קבלת מילת הקוד ה-j-ית ו-EC_j הסיביות הנוספות, המפענח פולט את סדרת הסימבולים המתאימים (אפסים ואחדים) ומשלים את טבלת התרגום. עדכון הטבלה נעשה עבור כל מילת קוד (כמו במקרה הקודם) ועבור כל אחת מהסיביות הנוספות שמלוות את מילת הקוד.

ניתוח יעילות המפענח:

זמן חיפוש של מילת קוד בטבלת התרגום נתון על ידי $O(1)$. זמן עידכון הטבלה נתון על ידי $O(EC_j+1)$ עבור מילת הקוד ה-j-ית.
כמות הזכרון הדרושה לאחסון טבלת התרגום שווה לזו שדרושה לבנית עץ הפיסוק במקדד; כלומר:

$$M_R \quad 2\left((p+1) + \sum_{j=1}^{p+1} EC_j\right).$$

2.3 השוואת ביצועם של המימושים השונים

ההשוואה מתבצעת בשני מישורים, הראשון הוא ניתוח אנליטי של יחס הדחיסה ושל כמות הזכרון הדרושה לביצוע הקדוד. במישור השני נציג תוצאות מספריות של יחס הדחיסה שהתקבל לאחר הפעלת המימושים השונים על תמונות אמיתיות.

2.3.1 ניתוח אנליטי

בסעיף זה נשווה ביצועם של שני המימושים הבאים:

1) פיסוק בעזרת עץ מדרגה α (סעיף מס' 2.2.1) של סדרה x_1^N כלשהי;

2) פיסוק בעזרת עץ בינרי של סדרה $\hat{x}_1^n$ (היצוג הבינרי של x_1^N) תוך גבולות ה-"Byte".

בשני ישומים אלה מתקבל אותו מספר של מחרוזות $(p+1)$ לאחר פיסוק הסדרה x_1^N ו- $\hat{x}_1^n$ בהתאמה.

הקצבת סיביות למילות הקוד:

מספר הסיביות הדרוש לקידוד $p+1$ המחרוזות בישום הראשון נתון על ידי:

$$L^T = \sum_{j=1}^{p+1} \lceil \log((\alpha-1)j) \rceil \qquad (2.5)$$

מספר הסיביות הדרוש בישום השני נתון על ידי:

$$L^B = \sum_{j=1}^{p+1} \{\lceil \log(j+\sum_{i=1}^{j-1} EC_i)\rceil + EC_j\} \qquad (2.9)$$

לצורך ההשוואה נניח ש- $\langle EC_j \rangle = k/2$ לכל j , כאשר $\langle\ \rangle$ מסמן ממוצע, ומתקבל (ראה נספח ו):

$$\log(2+\lceil\log\alpha\rceil) < 1 + \log((\alpha-1)/2)$$

מתוצאה זו רואים כי עבור $\alpha \geq 3$ מספר הסיביות הדרוש לקידוד הסדרה $\hat{x}_1^n$ קטן מזה הדרוש לקידוד הסדרה x_1^N , דהיינו $L^B < L^T$.

<u>כמות הזכרון הדרושה:</u>

כמות הזכרון הדרושה, במימוש המבוסס על עץ מדרגה α , נתונה על ידי:

$$M_R \propto (p+1)\alpha \qquad (2.7)$$

כמות הזכרון הדרושה בישום השני נתונה על ידי:

$$M_R^B \propto 2((p+1) + \sum_{j=1}^{p+1} EC_j) \qquad (2.10)$$

נניח שוב $\langle EC_j \rangle = k/2$, נציב ב-(2.10) ונשווה את שני הביטויים:

$$(p+1)\ (k+2) \lessgtr (p+1)\alpha \qquad (2.11)$$

$$\Rightarrow \lceil\log\ \alpha\rceil + 2 < \alpha \qquad \Rightarrow \quad M_R^B << M_R$$

עבור $\alpha \geq 4$.

מכאן נובע שבדרך כלל במימוש השני דרושה כמות זכרון פחותה מזו שבראשון.

<u>זמן יצירת מילת הקוד:</u>

זמן יצירת מילת הקוד נתון במימוש הראשון על ידי:

$$O(\ell_j + \alpha) \qquad (2.12)$$

ובמימוש השני על ידי:

$$O(\hat{\ell}_j + EC_j) \qquad (2.13)$$

כאשר $\hat{\ell}_j \simeq \ell_j \cdot k$; שוב נניח $\langle EC_j \rangle = k/2$, נציב ב-(2.13) ונקבל שזמן יצירת מילת הקוד במימוש השני נתון על ידי:

$$O(k(\ell_j + 1/2)) \approx O(k\ell_j) \qquad (2.14)$$

כדי לקבוע איזה מהמימושים מהיר יותר נתיחס לאי השוויון $k\ell_j \gtrless \ell_j+\alpha$,
כאשר $k = |\log \alpha|$. כיוון אי השוויון תלוי באורך המחרוזת ℓ_j. נסמן
ב-ℓ_j^* את אורך המחרוזת הנתון על ידי:

$$\ell_j^* \leq \left\lfloor \frac{-\alpha}{1-k} \right\rfloor$$

אשר קובע את כיוון אי השוויון. עבור אורכים ℓ_j, $\ell_j \leq \ell_j^*$ מתקבל
ש-$k\ell_j < \ell_j + \alpha$. לדוגמא עבור $\alpha = 256$ $(k = 8)$ נקבל $\ell_j^* = 36$.
משום שאין אפשרות בעת הקידוד לחזות את מספר המחרוזות מסך הכל המחרוזות
שאורכן קטן או גדול מ-ℓ_j^*, לא ניתן לקבוע באופן חד משמעי איזה מהמימושים
טוב יותר.
אם זאת, יש לציין שככל שהסדרה דחוסה יותר מתקבלות מחרוזות ארוכות יותר בתהליך
הפיסוק וקצב הפקת מילות הקוד גבוה יותר במימוש הראשון מאשר במימוש השני.

סיכום

תוצאות ההשוואה מוכיחות שמימוש המבוסס על עץ פיסוק בינרי תוך שמירת גבולות
ה-byte עדיף על מימוש המבוסס על עץ מדרגה α מהבחינות הבאות: הקצבת סיביות
למילות הקוד וכמות הזכרון הדרושה לקידוד.

2.3.2 ניתוח אנליטי

בסעיף זה נשווה את ביצועם של שני המימושים הבאים:

1) פיסוק בעזרת עץ בינרי רגיל.

2) פיסוק בעזרת עץ בינרי תוך שמירת גבולות ה-byte.

שני מימושים אלה יופעלו על סדרה בינרית $\hat{x}_1^n$ המיצגת סידרה x_1^N כלשהיא מעל A
כאשר ל-A עוצמה α כלשהיא.

הקצבת סיביות למילת קוד

עבור סדרה נתונה, מספר המחרוזות המתקבל בכל אחד מהמימושים שונה, לכן אין אפשרות
להשוואה אנליטית ביניהם. במימוש (1) לא מנצלים את הקורלציה בין אותיות הא"ב
המקורי, לכן צפוי שתוצאות הדחיסה שלו תהיינה פחות טובות מאלה של מימוש (2).

<u>כמות זכרון דרושה:</u>

כמות הזכרון הדרושה במימוש הראשון נתונה על ידי הביטוי:

$$M_R^b \propto 2(p_b+1) \qquad (2.15)$$

ובישום השני על ידי הביטוי (2.10):

$$M_R^B \propto 2\left((p_B+1) + \sum_{j=1}^{p_B+1} EC_j\right)$$

לא ניתן לקבוע איזה מהביטויים גדול יותר מפני שאין קשר הדוק וידוע בין p_b ל-$\sum_{j=1}^{p_B+1} EC_j + p_B$.

<u>זמן יצירת מילת הקוד</u>

זמן יצירת מילת הקוד בישום הראשון נתון על ידי :

$$O(\ell_j^b + 2) \qquad (2.16)$$

כאשר ℓ_j הוא אורך המחרוזת ה-j-ית.

במימוש השני, זמן יצירת מילת הקוד נתון על ידי:

$$O(\ell_j^B + EC_j) \qquad (2.17)$$

כאשר $\ell_j^B \triangleq \ell_j k$ $\quad k = \lceil \log \alpha \rceil \quad \ell_j = 0,1,2 \ldots$ ושוב נניח $<EC_j> = k/2$.

מכאן שניתן לרשום את (2.17) בצורה הבאה:

$$O(\ell_j k) \qquad (2.18)$$

כאשר $\ell_j = 1,2,\ldots$. בדרך כלל מתקיים $\ell_j k > \ell_j^b + 2$, ולכן המימוש הראשון מהיר יותר.

<u>המפענחים:</u>

המפענח של המימוש הראשון הוא בעל סיבוכיות נמוכה מזו של המימוש השני. עובדה זו נובעת מהצורך של המפענח במימוש השני להתיחס למילות הקוד וגם לסיביות הנוספות הנלוות אליה. זמן חיפוש מילת הקוד בשני המקרים נתון על ידי O(1), זמן עידכון הטבלה במימוש הראשון הינו קבוע O(1), ובישום השני הוא תלוי ב-EC_j (מספר הסיביות הנוספות) דהיינו $O(EC_j)$.

סיכום:

סיבוכיות המימוש הראשון במקדד ובמפענח נמוכה יותר מזו של המימוש השני.
לא נמצאה אפשרות להשוואה אנליטית של יכולת הדחיסה של 2 המימושים.

2.3.3 תוצאות מספריות

בסעיף זה נציג את מספרי הסיביות לפיקסל שהתקבלו עבור מימושים שונים על תמונות פנים המיוצגים על ידי 4 סיביות לפיקסל. המימושים הם:

1) קידוד בעזרת עץ בינרי רגיל,

2) קידוד בעזרת עץ בינרי תוך שמירת גבולות ה-Byte.

נסמן ב-$\rho(u)$ את מספר הסיביות לפיקסל המתקבל לאחר הדחיסה וב-$\hat{H}(u)$ את משערך האנטרופיה שהוא גם החסם התחתון לדחיסה על האלגוריתם ZL.
$\hat{H}(u)$ נמדד ביחידות של סיביות לפיקסל.

התוצאות שמתקבלות עבור תמונות המיוצגות על ידי 4 סיביות מתוארות בטבלה מס' 2.

טבלה מס' 2.2: מספרי הסיביות לפיקסל המתקבלים על ידי המימושים: 1) עץ בינרי רגיל, 2) עץ בינרי תוך שמירת גבולות ה-Byte.

Table 2.2: $\rho(u)$ (bits/pixel) obtained by 1) Binary Parsing Tree, 2) Binary Parsing Tree with Byte Limiter.

תמונה	$\hat{H}(u)$	(מימוש 1) $\rho(u)$	(מימוש 2) $\rho(u)$	($\frac{\text{סיביות}}{\text{פיקסל}}$)
1	1.3854	2.2078	1.8206	
2	1.5192	2.3989	1.9890	
3	1.2525	1.9990	1.6444	
ממוצע	1.3857	2.2019	1.8180	

התוצאות הנ"ל מאשרות את הטענה שתוצאות הדחיסה של המימוש השני טובות יותר, וזאת מכיוון שמספר הסיביות לפיקסל המתקבל על ידי מימוש 2 נמוך יותר ולכן קרוב יותר ל-$\hat{H}(u)$ מזה שמתקבל על ידי מימוש 1.

פרק 3 : הרחבות של אלגוריתם זיו-למפל

3.1 מבוא

בפרק זה נציע ונדון בשלוש גרסאות של מימוש אלגוריתם זיו-למפל, שמטרתן לשפר את ביצועיו.

למימושים שהוצגו עד כה מספר חסרונות:

1) אורך משתנה של מילות הקוד.

2) ייצוג מסורבל כיוון שיציאת המקדד מורכבת ממלות קוד ומסיביות נוספות הנלוות אליה.

3) משך זמן ארוך נדרש לקידוד ולפיענוח של תמונות. תכונה זו פוגעת בביצועי האלגוריתם כאשר משתמשים בו לאגירת אינפורמציה, וכאשר נדרשת שליפה מהירה של תמונות בחיפוש אחר אינפורמציה ויזואלית.

4) האלגוריתם אינו חסין לרעש, לכן,כדי להשתמש בו למטרות תקשורת יש צורך בסינכרון ובהגנה מפני שגיאות ערוץ.

5) דרושה כמות זכרון גדולה לקידוד ולפיענוח.

התגברות משמעותית על חסרונות אלה מתאפשרת באמצעות שתי גרסאות של האלגוריתם המוצעות כאן.

גרסה אחת תביא לשפור החסרונות שהוצגו בסעיפים 2, 3, 5, ובנוסף תאפשר גם שפור בדחיסת המידע בהשוואה לדחיסה המושגת על ידי מימושים שונים אחרים. בגרסה זו נממש את האלגוריתם בכל מישור סיבית בנפרד והיא תקרא BP-ZL (Bit Plane - Ziv Lempel).

הגרסה השניה, המבוססת על שמוש בעץ פיסוק אוניברסלי (Universal-Tree), תביא לשיפור החסרונות שהוצגו בסעיפים 1, 2 ו-4. באמצעות גירסה זו מתקבלות מילות קוד בעלות אורך קבוע, וקיים שיפור משמעותי בחסינות מפני רעש מבלי לפגוע בדחיסה שהושגה על ידי הגירסה המקורית של האלגוריתם. המימוש של הגרסה השניה על ידי פיסוק בינרי תוך שמירת גבולות ה-Byte יקרא UT-ZL.

על מנת לשפר את כל 5 החסרונות נשתמש בגירסה משופרת שלישית של אלגוריתם ZL אשר משלבת את שתי הגרסאות הקודמות, BP-ZL ועץ פיסוק אוניברסלי.
בהמשך נתאר את שלושת הגרסאות וממושן ונציג את תוצאות הדחיסה המתקבלות באמצעותן.

3.2 ממוש האלגוריתם במישורי סיבית נפרדים

שמו של המימוש נובע מהדרך בה מפעילים את אלגוריתם זיו-למפל על התמונה. במקרים בהם התמונה מוצגת על ידי 4 סיביות לפיקסל, המימוש נקרא 4BP-ZL (4 מישורי סיבית).

ערכי הפיקסלים של תמונה המיוצגת על ידי 4 סיביות לפיקסל שייכים לא"ב Y המכיל 16 סימבולים שונים ($\alpha = 16$), $Y = \{0,16,\ldots,240\}$.
ערך של פיקסל y כלשהוא, $y \in Y$ נתון על ידי:

$$y = \sum_{i=0}^{3} 2^{(7-i)} a_i \qquad a_i \in \{0,1\}$$

כאשר היצוג הבינרי של y הינו: $y = a_0\ a_1\ a_2\ a_3$.
a_0 היא הסיבית המשמעותית ביותר (MSB) ו-a_3 היא הסיבית הפחות משמעותית ביותר (LSB).

<u>הגדרה:</u> מישור סיבית (bit plane) הינו אוסף של סיביות בעלות אותו משקל ביצוג ערכי הפיקסלים. נסמן מישור סיבית על ידי האות ℓ.
לדוגמא, מישור הסיבית המשמעותית ביותר $\ell = 0$ של תמונה הוא אוסף כל הסיביות a_0 של ערכי הפיקסלים המייצגים את התמונה.

<u>אופן הקידוד:</u> הקידוד נעשה על כל מישור סיבית לחוד, כלומר מפעילים את האלגוריתם זיו-למפל במקביל על 4 סדרות סימבולים בינרים כאשר כל סדרה מתאימה למישור סיבית אחר. כדי לנצל את הקורלציה הקיימת בין פיקסלים סמוכים הקדוד מתבצע על מישורי הסיביות השייכות ליצוג GRAY של ערכי הפיקסלים.

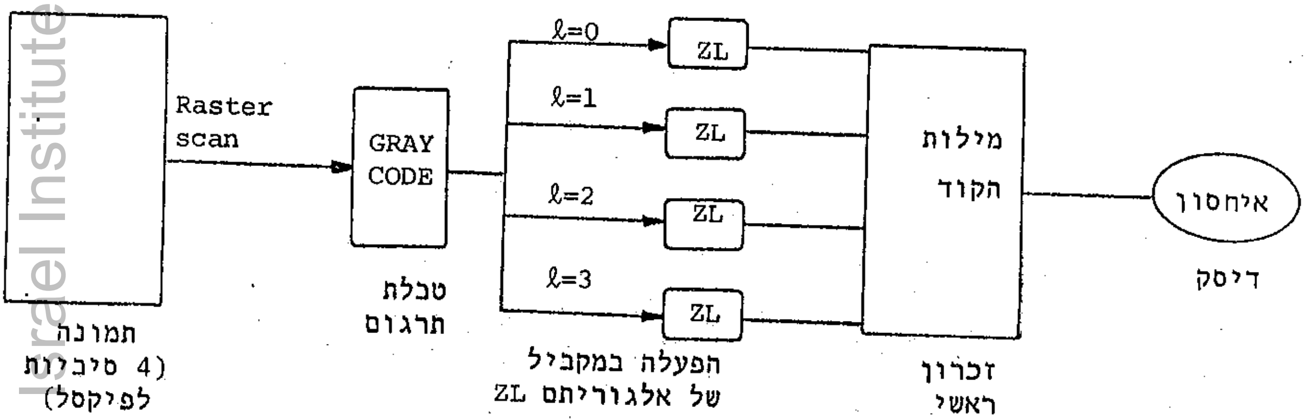
מדוע יצוג GRAY?: קידוד של כל מישור סיבית לחוד פוגע בקורלציה הקיימת בין פיקסלים סמוכים, משום שביצוג הבינרי, שני פיקסלים הקרובים בערכם המספרי הם לא בהכרח קרובים זה לזה לפי מרחק Hamming. מרחק Hamming של שני ערכים מספריים עוקבים ביצוג GRAY שווה ל-1, ולכן הקורלציה בין ערכי פיקסלים סמוכים תשמר במידה גדולה יותר בכל מישור סיבית ביצוג GRAY. הסבר זה ניתן לבדיקה על ידי מודל ליצירת תמונה אשר מוכיח את יתרונות יצוג GRAY של תמונות למטרות קידוד. בעזרת אותו מודל ניתן להעריך גם את ערכי הדחיסה המתקבלים על ידי מימוש זה.
תאור וניתוח המודל נמצא בסעיף 4.1 בהמשך העבודה.

3.2.1 <u>תאור סכמת הקידוד 4BP-ZL</u>

קיימות מס' אפשרויות לקידוד במימוש 4BP-ZL. האלגוריתם זהה בכל המקרים וההבדל בין מימוש למימוש תלוי בדרגת השכלול של המערכת המבצעת את הקידוד.

1. <u>מימוש 4BP-ZL בזמן אמיתי</u>

דיאגרמת הבלוקים של המערכת:



<u>ציור מס' 3.1</u>: דיאגרמת בלוקים של המימוש 4BP-ZL (ביצוע קידוד בזמן אמיתי).
<u>Fig. 4.1</u>: Block diagram of the 4BP-ZL implementation (ON-LINE).

שלבי הקידוד:

א. תירגום ערכי הפיקסלים ליצוגם בקוד GRAY נעשה על ידי טבלת תרגום שנמצאת ביחידה A/D.

ב. קידוד סדרות הסיביות בארבעת המישורים נעשה על ידי 4 יחידות מקבילות אשר כל אחת מהן מבצעת את הקידוד בעזרת עץ בינרי רגיל כפי שתואר בסעיף 2.2.3.

ג. מילות הקוד מועברות לזכרון הראשי של המערכת או של המחשב, כאשר לכל מישור סיבית יש איזור נפרד לאחסון זמני של מילות הקוד שלה.

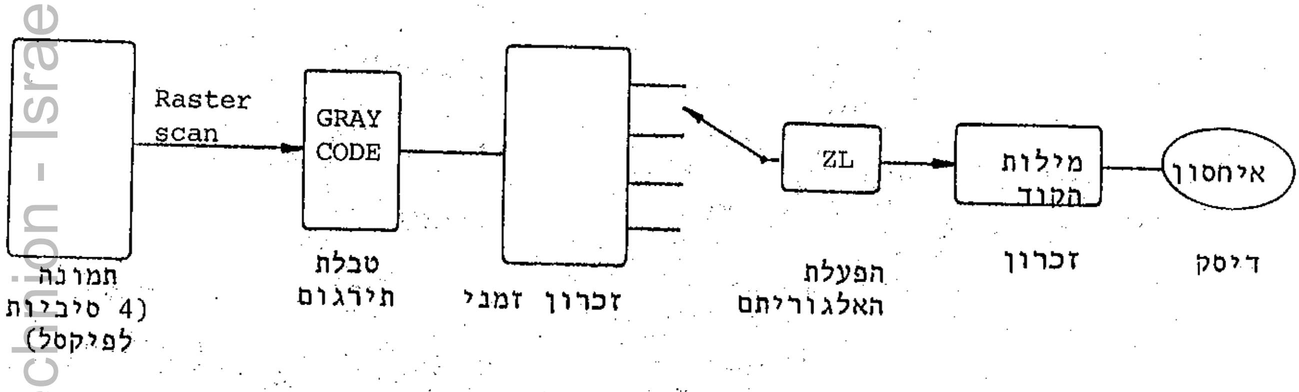
ד. בסיום תהליך הקידוד מועברות מילות הקוד בצורה מהירה לאיחסון בדיסק או בסרט.

הקידוד מסתיים באופן הדרגתי. תחילה מועברות מילות הקוד של מישור הסיבית ה-MSB ($\ell = 0$), ובו זמנית ממשיך להתבצע קידוד במישורי הסיביות האחרים. סיום הדרגתי של קידוד מישורי הסיבית מאפשר ניצול טוב יותר של זמן ההעברה לדיסק אשר מאיץ את תהליך הקידוד.

.2 <u>מימוש 4BP-ZL ב-OFF-LINE</u>

בביצוע זה לא משתמשים במערכת מיוחדת. ניתן לבצע את תהליך הקידוד בעזרת מחשב רגיל, אך לא בזמן אמיתי.

דיאגרמת הבלוקים של המערכת:



<u>ציור מס' 3.2</u>: דיאגרמת בלוקים של המימוש 4BP-ZL (ביצוע ב-OFF-LINE).

<u>Fig. 4.2:</u> Block diagram of the 4BP-ZL implementation (OFF-LINE).

שלבי הקידוד:

א. תרגום ערכי הפיקסלים ליצוגם לפי קוד GRAY נעשה על ידי טבלת תירגום.

ב. התמונה בשלמותה מאוחסנת בזכרון זמני לפני ביצוע הקידוד.

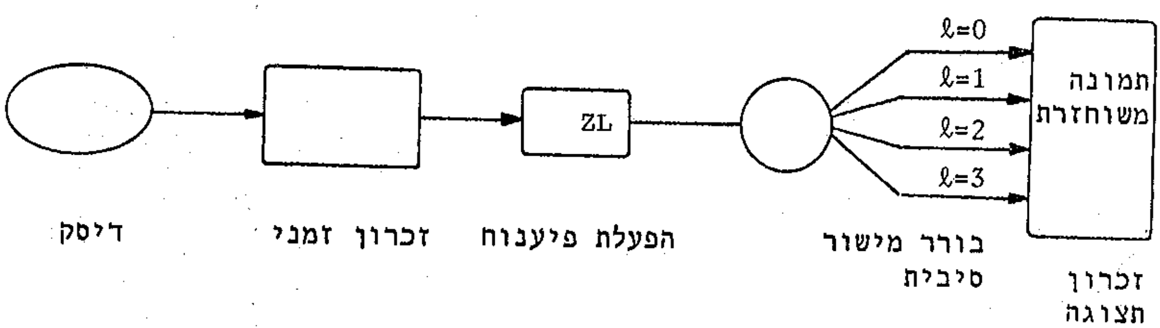
ג. תהליך הקידוד נעשה באופן טורי לכל מישור סיבית. התהליך מתחיל במישור הסיבית המשמעותית ביותר ומסתיים במישור הסיבית הפחות משמעותית ביותר (LSB).

ד. איחסון בדיסק נעשה עם סיום הקידוד של כל רמה סיבית.

3.2.2 תאור סכמת הפיענוח 4BP-ZL

רצוי שמערכת הפיענוח תהיה פשוטה ככל האפשר כדי לא לחייב את המשתמש להתקין חומרה מיוחדת.

דיאגרמת הבלוקים של מערכת הפיענוח:



ציור מס' 3.3: דיאגרמת בלוקים של המפענח של המימוש 4BP-ZL.

Fig. 4.3: Block diagram for the 4BP-ZL implementation decoder.

שלבי הפענוח:

עבור כל אחד מארבעת מישורי הסיביות מבצעים את שני השלבים הבאים:

א. כל מילות הקוד השייכות לאותו מישור סיבית מועברות מהדיסק לזכרון הזמני.

ב. כל סיבית המתקבלת באמצעות תהליך הפיענוח מועברת לזכרון תצוגה במישור המתאים.

כאשר אין אפשרות לפנות לכל מישור סיבית לחוד בזכרון התצוגה, העדכון ההדרגתי של התמונה נעשה על ידי סיכום ערך הפיקסל הנמצא בזכרון התצוגה עם הסיבית המתקבלת מהפיענוח (במשקל מתאים).

פיענוח מילות הקוד הוא מידי. זמן הפענוח נתון על ידי $O(1)$, ראה סעיף 2.2.1.

3.2.3 יתרון פענוח הדרגתי של תמונה

לאחר הפענוח של שני מישורי הסיביות המשמעותיות ביותר, ולעיתים מספקת הסיבית היותר משמעותית בלבד, ניתן להבחין באופן ברור בתמונה, על כן כבר בשלב זה ניתן להפסיק את פענוחם של המישורים שנותרו אם מסתבר שהתמונה אינה מעניינת לתצוגה.

בהתחשב בכך שבמערכות Photovideotex קצב העברת הנתונים הוא 9600 bits/sec, והדחיסה הממוצעת של 2 מישורי הסיבית המשמעותית ביותר ($\ell = 0$ ו-$\ell = 1$), כפי שהתקבל מתוצאות הדחיסה של 3 תמונות פנים, הוא של כ-0.25 bit/pixel עבור כל מישור סיבית, נווכח שזמן השחזור של התמונה המספיק כדי להבחין בה הוא:

$$\frac{256^2 \times 0.25 \times 2}{9600} = 3.4 \text{ sec}$$

ולעיתים אף מחצית הזמן (1.7 sec) אם הסיבית ה-MSB מספיקה להבחנה.

במימוש שמתחשב בגבולות ה-byte שהוצג בסעיף 2.2.4 מתקבל לאחר דחיסה שהמספר הממוצע של סיביות לפיקסל הוא: 1.8 bit/pixel, לכן, עבור קצב העברת הנתונים של 9600 bits/sec הזמן הדרוש לשחזור תמונה שלמה הוא כ-12 שניות. מכאן שעבור חיפוש תמונות במאגר נתונים השיטה 4BP-ZL מאיצה את החיפוש עד כדי פי 7.

בנספח י' ניתן לראות הדגמה של שיחזור הדרגתי של תמונה מס' 1.

3.2.4 ניתוח המימוש 4BP-ZL

הקידוד והפיענוח בעזרת המימוש 4BP-ZL של האלגוריתם זיו-למפל שקול ל-4 הפעולות של האלגוריתם הממומש על ידי עץ פיסוק בינרי רגיל כפי שהוצג בסעיף 2.2.3. מכאן שבניתוח יעילות המימוש מתקבלות אותן התוצאות שהתקבלו בניתוח של המימוש המבוסס על עץ פיסוק בינרי רגיל מבחינת הקצבת סיביות למילות הקוד, כמות הזכרון הדרושה לעץ הפיסוק, זמן החיפוש בעץ וזמן הפיענוח והעידכון של טבלת הפענוח. למרות זאת במימוש זה נצטרך להוסיף בניתוח הסיבוכיות את תרומתן של הפעולות הנוספות הדרושות להפעלה החוזרת של האלגוריתם לכל מישור סיבית.

להלן תאור פעולות אלה:

בקידוד:

מימוש 1 (ON LINE): יש לנתב את זרימת מילות הקוד לאיזור המתאים בזכרון. זוהי פעולה מידית הכרוכה בעידכון מחוון הזכרון המתאים. כמות הזכרון הנוספת הדרושה לאיחסון מילות הקוד עד שהן מועברות לדיסק שווה לכמות הזכרון הדרושה לאיחסון תמונה שלמה (מכיוון שיש להתחשב במקרה בו התמונה לא ניתנת לדחיסה).

מימוש 2 (OFF LINE): יש לנתב את קריאת הסיביות ממישור הסיבית המתאים כדי לקדדן. איחסון זמני של מילות הקוד אינו דורש טיפול מיוחד. כמות הזכרון הנוספת הדרושה לאיחסון התמונה העוברת קידוד ולמילות הקוד של מישור סיבית אחד שווה לכמות הזכרון הדרושה לאיחסון תמונה ורבע.

בפיענוח: הפעולות הנוספות הדרושות תהיינה תלויות במבנה זכרון התצוגה של המערכת. עבור מערכות בהן ניתן לפנות לכל סיבית לחוד דרושה רק פניה למישור הסיבית המתאים. עבור מערכות בהן לא ניתן לפנות לכל מישור סיבית, יש לקרוא מזכרון התצוגה ולבצע סיכום עבור כל פיקסל ועבור כל מישור סיבית פרט למישור $\ell = 0$. כמות הזכרון הנוספת הדרושה לאיחסון מילות הקוד של מישור סיבית אחד שווה לכמות הזכרון הדרושה לאיחסון רבע תמונה.

3.2.5 השוואת המימושים השונים של אלגוריתם זיו-למפל

בטבלה מס' 1 מופיעים ערכי $\rho(u)$ (מספר הסיביות לפיקסל המתקבל לאחר דחיסה) שמתקבלים עבור 3 תמונות פנים בשני מימושים של האלגוריתם זיו-למפל. 1) קידוד של היצוג בקוד GRAY של התמונה (4 מישורי סיבית), 2) קידוד בעזרת עץ בינרי תוך שמירת גבולות ה-byte (4 הסיביות המשמעותיות ביותר).

טבלה מס' 3.1: מספר סיביות לפיקסל לאחר הדחיסה המתקבל על ידי שני המימושים של האלגוריתם זיו-למפל.

Table 4.1: No. of bits / pixel received by two implementations of the ZL. algorithm.

תמונה	$\hat{H}(u)$	$\rho_{4BP}(u)$ (מימוש 1)	$\rho(u)$ (מימוש 2) $(\frac{\text{סיביות}}{\text{פיקסל}})$
1	1.3854	1.7337	1.8206
2	1.5192	1.6811	1.9890
3	1.2525	1.4348	1.6444
ממוצע	1.3857	1.6166	1.8180

בטבלה מס' 3.2 מופיע מספר המחרוזות p שנוצרות בתהליך הפיסוק של תמונות בגודל 256×256 בשני המימושים. של קידוד ZL.

טבלה מס' 3.2: מספר המחרוזות שנוצרות בקידוד ZL (תמונות בגודל 256×256).

Table 4.2: No. of strings received after parsing 256x256 pictures with the ZL algorithm.

תמונה	$p(\hat{H}(u))^*$	מימוש 1	מימוש 2
1	7640	10824	7640
2	8282	10494	8282
3	6944	9113	6944
ממוצע	7622	10824	7622

* מס' המחרוזות המתקבל בחישוב $\hat{H}(u)$.

מתוך טבלאות הנתונים הנ"ל ניתן לקבוע שבשימוש 4BP-ZL של האלגוריתם מתקבלות תוצאות דחיסה טובות יותר וזאת משום שמספר הסיביות המתקבל לאחר הדחיסה נמוך יותר וקרוב יותר לערכי $\hat{H}(u)$ מאשר אלה שבשימוש השני, למרות, שבשימוש 4BP-ZL מספר המחרוזות גדול יותר, אך אורכי מילות הקוד קטנים יותר מפני שמספר המחרוזות מתקבל מ-4 עצים שונים; מכאן שיחס הדחיסה של המימוש 4BP-ZL טוב יותר.

3.3 ממוש האלגוריתם על ידי עץ פסוק אוניברסלי

ניתן לממש את אלגוריתם זיו-למפל בעזרת עץ פיסוק קבוע ונתון מראש, הנקרא עץ אוניברסלי והמוגדר להלן:

הגדרה: עץ אוניברסלי הינו עץ פיסוק שנוצר בתהליך הפיסוק של סדרה אופינית (סדרה המיצגת קבוצת סדרות), והוא מיועד לשמש כעץ פיסוק לקידוד ולפיענוח של סדרות סימבולים בקבוצה אותה הוא מייצג.
מכאן יחודיות הגירסה בהשוואה לאלגוריתם המקורי אשר מבוסס על עץ פיסוק יחודי עבור כל סדרה.

אופן הקידוד: תהליך הקידוד מורכב משני שלבים:
בשלב הראשון נבנה עץ פיסוק אוניברסלי המיצג את התכונות הסטטיסטיות של קבוצת סדרות מסוימת. ביצוע חוזר של שלב זה, לאחר הפעם הראשונה, נעשה רק כאשר העץ הקיים לא מתאים לתכונות הסטטיסטיות של התמונות שצריכות לעבור קידוד.

בשלב השני מתבצע הקידוד עצמו של התמונות לפי עץ הפיסוק הבנוי מראש.

אופן הפיענוח: פיענוח התמונות נעשה על פי אותו עץ אוניברסלי ששימש כעץ פיסוק בתהליך הקידוד.
שני שלבי הקידוד ושלב הפיענוח ניתנים לביצוע על ידי המימושים השונים של האלגוריתם שתוארו עד כה.

תכונות גירסת עץ אוניברסלי:

א) שימוש בעץ פיסוק אוניברסלי מאפשר קבלת מילות קוד באורך קבוע. אורך מילות הקוד נקבע על פי גודל העץ, זאת אומרת על פי מספר העלים בעץ.

ב) שימוש בעץ אוניברסלי מפשט את תהליך פליטת מילות הקוד מפני שאין צורך בעידכון העץ ולא בקביעת אורך של כל מילת קוד לחוד. הפישוט מורגש בעיקר במימוש המבוסס על עץ פיסוק בינרי תוך שמירת גבולות ה-Byte מפני שבו אין צורך בשידור סיביות נוספות, פרט למילות הקוד כפי שנעשה במימוש ללא עץ אוניברסלי.

ג) מימוש זה מאפשר שימוש באלגוריתם ZL למטרות תקשורת מכיוון שעץ הקדוד והפיענוח ידוע מראש למשדר ולמקלט ועל ידי כך עולה החסינות לרעש. כדי לשפר את ביצועי האלגוריתם בשמושי תקשורת יש להבטיח שלשגיאות הערוץ תהיינה השפעות מקומיות בלבד, וזאת על ידי הוספת סימבולי סינכרון לבלוקים של פיקסלים. אורך הבלוק נקבע על ידי מתכנן המערכת; לדוגמא, ניתן להחליט שאורך הבלוק יהיה כאורך שורה אחת בתמונה.

3.3.1 סכמת הקידוד UT-ZL

המימוש ה"טבעי" של האלגוריתם הינו המימוש המבוסס על קידוד בעזרת עץ פיסוק בינרי תוך שמירת גבולות ה-Byte, אותו נתאר בהמשך. לצורך פישוט הסימונים נסמנו באותיות UT-ZL.

בתאור שלבי האלגוריתם נתיחס לתמונה כאל סדרת סימבולים שניתן לראותה בשני אופנים, האחד כסדרה בעלת N סימבולים השייכים לא"ב של 2^k סימבולים והשני כסדרה בעלת $N \cdot k$ סימבולים בינריים. k הינו מספר הסיביות המייצגות סימבול באלפבית המקורי.

תהליך הקידוד מורכב משני שלבים:

שלב I: בנית העץ האוניברסלי

נתאר כעת את תהליך בנית העץ המייצג את התכונות הסטטיסטיות של קבוצת סדרות מסויימת והוא כדלקמן:

א. יצירת עץ בינרי התחלתי המורכב מ-2^k עלים, כאשר k הוא מספר הסיביות המייצגות סימבול. העלים של העץ ההתחלתי מיצגים את כל הצרופים האפשריים הניתנים להשגה על ידי איחוד של k סיביות, זאת אומרת את 2^k הסימבולים של האלפבית המקורי. במקרה בו $k = 4$ העץ ההתחלתי מורכב מ-16 עלים.

ב. הפעלת אלגוריתם ZL, שלבים 1 עד 5 של מימוש עץ פיסוק בינרי תוך שמירת גבולות ה-Byte (ראה סעיף 2.2.4) על הסדרה "האופיינית".
בתהליך קידודה של הסדרה האופיינית נפלטות מילות קוד ונבנה עץ הפיסוק אשר ישמש כעץ פיסוק אוניברסלי לסדרות הבאות.

ג. איחסון העץ בזכרון המחשב או העברתו למקלט במערכות תקשורת כדי להשתמש בו לקידוד ופענוח הסדרות הבאות.

<u>שלב II: קידוד הסדרות</u>

קיימות שתי שיטות למימוש שלב זה. האחת קידוד ללא גרימת עוות והשניה קידוד תוך גרימת עוות. בגירסה השניה הקידוד איננו מתאר את הנתונים באופן מלא ולכן בעת פענוח מילות הקוד יש להשלים את הסיביות החסרות, דבר הגורם לעוותים.

<u>שיטה א' - קידוד ללא גרימת עוות</u>

שלבי האלגוריתם הם:

(0) החל מהמקום הנוכחי של המחוון בסדרת הכניסה, מצא את המחרוזת הארוכה ביותר אשר שייכת לעץ הפיסוק.

(1) פלוט את הקוד של המספר המזהה של המחרוזת שנמצאה, דהיינו את המספר המזהה של העלה.

(2) אם המחוון בסדרת הכניסה <u>מצביע</u> על סיום של ה-Byte. אזי :

A) קדם את המחוון לתחילת המחרוזת החדשה,

B) חזור לשלב (0) (אם המחוון לא מצביע על סוף סדרת הנתונים),

(3) אם המחוון בסדרת הכניסה <u>איננו מצביע</u> על סיום של ה-Byte, אזי:

A) החזר את המחוון למקום המצביע על תחילת ה-Byte הנוכחי.

B) חזור לשלב (0) (אם המחוון לא מצביע על סוף סדרת הנתונים).

שיטה ב׳: קידוד תוך גרימת עוות

שלבי האלגוריתם הם:

(0) זהה לשיטה א׳,

(1) זהה לשיטה א׳,

(2) זהה לשיטה א׳,

(3) אם המחוון בסדרת הכניסה איננו מצביע על סיום של ה-Byte אזי:

A) קדם את המחוון עד למקום המצביע על התחלת ה-Byte הבא.

B) חזור לשלב (0) (אם המחוון לא מצביע על סוף סדרת הנתונים).

ההבדל בין שתי השיטות לקידוד הוא בכך שבשיטה הראשונה המחוון מוחזר לתחילת ה-Byte הנוכחי ובשניה המחוון מתקדם לתחילת ה-Byte הבא ועל ידי כך מדלג על קידוד מספר סיביות.

דוגמא

נדגים את האלגוריתם בעזרת הסדרות x ו- y מעל $A = \{a,b,c\}, A$ $\alpha = |A| = 3$.

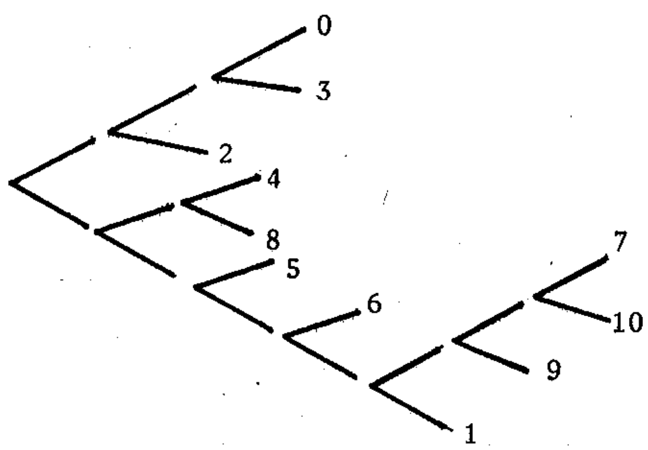
שלב I: בנית העץ

תהיה נתונה הסדרה x (סדרה אופינית) שהופיעה בדוגמא שבעמוד 21 .

$x_1^N =$ a b b b c b b a

$\hat{x}_1^n = \lambda$ 00 11 11 11 10 11 11 00

עץ הפיסוק האוניברסלי שמתקבל הינו:



ציור 3.4: עץ אוניברסלי של הדוגמא שנוצר בשלב I (שלב בנית העץ).

Fig. 3.4: Universal parsing tree for the example build in step I .

העץ בעל 11 עלים, לכן אורך מילות הקוד הינו $4 = \lceil \log 11 \rceil$.

<u>שלב II</u>: קידוד הסדרה

תהיה נתונה הסדרה הבאה:

$$Y_1^N = \quad b \quad a \quad c \quad c \quad b \quad b \quad a \quad b \quad c \; ....$$

$$y_1^n = \quad 11\;00\;10\;10\;11\;11\;00\;11\;10\; ....$$

<u>שיטה א'</u>

לאחר הקידוד נקבל:

1) פיסוק הסדרה:

$$\underset{\uparrow}{1}1\;\underset{\uparrow}{0},0\;\underset{\uparrow}{1},0\;\underset{\uparrow}{1},0\;\underset{\uparrow}{1},1\;11\;00\;\underset{\uparrow}{1},1\;10,\uparrow$$

2) מילות הקוד: 5 3 8 8 10 6

כאשר $j \overset{\Delta}{=}$ מקום בו מפסקים את הסדרה (מגיעים לעלה בעץ הפיסוק האוניברסלי).

$\uparrow$ = מקום אליו המחוון חוזר (התחלת ה-Byte) כדי להמשיך את פיסוק הסימבולים הבאים.

<u>שיטה ב':</u>

לאחר הקידוד נקבל:

1) פיסוק הסדרה:

$$\underset{\downarrow}{1}1\;0,0\;\underset{\downarrow}{1}0\;1,0\;\underset{\downarrow}{1}1\;11\;00\;1,1\;\underset{\downarrow}{1}0$$

2) מילות הקוד: 5 8 10

כאשר, $, \overset{\Delta}{=}$ מקום בו מפסקים את הסדרה.

$\downarrow \overset{\Delta}{=}$ מקום אליו המחוון מתקדם כדי להמשיך את פיסוק הסימבולים הבאים.

<u>שיטה ג׳</u>: בשיטה זו, המבוססת על שיטה ב׳, לא נגרם עוות למידע העובר קידוד, מכיון שבנוסף למילות הקוד נפלטות גם הסיביות שלא הוקצבה להן מילת קוד. לכן, מתקבלות מילות קוד באורך לא קבוע. לא נדון בעבודה זו בשיטה ג׳.

3.3.2 <u>סכמת הפיענוח UT-ZL</u>

בתהליך הפיענוח משתמשים בעץ הפיסוק האוניברסלי שנוצר בשלב ה-I של הקידוד על מנת להתאים לכל מילת קוד שמתקבלת בעת הפיענוח את סדרת הסיביות השייכת לה.

כדי לפשט את תאור שלבי הפענוח נשתמש בסימונים הבאים:

$\ell c \overset{\Delta}{=}$ אורך סדרת הסיביות המתאימה למילת הקוד.

$mod\ (\ell c,k) \overset{\Delta}{=}$ פעולת מודולו k, כאשר k הוא מס׳ הסיביות ל-"Byte".

<u>שיטה א׳</u>: ללא טעות

(0) קרא למילת קוד.

(1) פלוט סדרת סיביות באורך $\ell t = \ell c - mod\ (\ell c,k)$ המתאימה למילת הקוד שנקראה.

(2) חזור לשלב (0).

יש לציין שפליטת סדרת סיביות באורך $\ell t = \ell c - mod\ (\ell c,k)$ שקולה לפעולה של החזרת המתוון להתחלת Byte בתהליך הקידוד.

<u>שיטה ב׳</u>: עם טעות

(0) קרא למילת קוד.

(1) פלוט את סדרת הסיביות המתאימה למילת הקוד שהתקבלה.

(2) אם $mod\ (\ell c,k) \neq 0$ $b\ell \overset{\Delta}{=}$ אזי הוסף $(k-b\ell)$ סיביות ל-ℓc שכבר נפלטו.

(3) חזור לשלב (0).

הוספת הסיביות שקולה לקדום המחוון בסדרת הכניסה בתהליך הקידוד. אופן בנית עץ הפיסוק מבטיח שהסיבית המשמעותית ביותר של ה-Byte תשוחזר במדויק. שיחזור הסיביות החסרות נתון לשיקול דעתו של מתכנן המערכת.

דוגמא: בפענוח הסדרות נשתמש בעץ האוניברסלי שהתקבל בדוגמא הקודמת. הדגמה של שיטה א': מילות הקוד שמתקבלות הן: 6 10 8 8 3 5. למילת הקוד 5 שייכת סדרת הסיביות " 0 1 1 " (ראה עץ אוניברסלי עמוד 39). לסדרה 3 סיביות, לכן $\ell c = 3$, $mod (\ell c,2) = 1$ ו-$\ell t = 3-1 = 2$; מכאן שרק 2 הסיביות הראשונות " 1 1 " נפלטות מהמפענח עבור מילת הקוד 5. תהליך זה חוזר עבור כל אחת ממילות הקוד שמתקבלות במפענח. לאחר פענוח מילות הקוד מתקבלת מחדש סדרת הסיביות המקורית $\hat{Y}_1^n$:
" 10 11 00 11 11 10 10 00 11".

הדגמה של שיטה ב': מילות הקוד שמתקבלות בתהליך הקידוד הן: 10, 8, 5. למילת הקוד 5 שייכת סדרת הסיביות " 0 1 1 " (הקידוד נעשה בעזרת העץ האוניברסלי שבדוגמא הקודמת). לסדרה 3 סיביות, לכן $b\ell = mod (3,2) = 1$ ויש צורך להוסיף $k - b\ell = 1$ סיבית ל-3 הסיביות השייכות למילת הקוד שפוענחה. נקבל ביציאת המפענח Ø 0 1 1, כאשר Ø מסמן את הסיבית '0' או '1' שנבחרה להשלמת הסדרה. הסדרה המתקבלת לאחר פיענוח מילות הקוד היא: 1Ø 00 11 11 1Ø 10 0Ø 11.

3.3.3 ניתוח המימוש UT-ZL

בסעיף זה נציג ניתוח אנליטי של ביצועי המימוש UT-ZL. מידת ההצלחה של מימוש זה נקבעה על סמך תוצאות הדחיסה שמתקבלות באמצעותו. הצגת התוצאות המספריות והשוואתן לתוצאות המתקבלות במימושים אחרים נמצאת בסעיף הבא.

קידוד

שלב I: שלב זה זהה מכל הבחינות למימוש של האלגוריתם בעזרת עץ פיסוק בינרי תוך שמירת גבולות ה-Byte. ראה סעיף 2.2.2.

כמות הסיביות בזכרון הדרושה לאיחסון העץ נתונה על ידי הביטוי:

$$M_R \propto K2((p+1) + \sum_{j=1}^{p+1} EC_j) \qquad (3.1)$$

כאשר p : מס' המחרוזות שנוצרות בתהליך הפיסוק של הסדרה האופיינית (שלב I),

EC_j : מס' הסיביות הנוספות למילת הקוד ה-j-ית.

וכאשר K הוא מקדם המבטא את אורך המחוונים (בסיביות) המצביעים על הצמתים שבעץ ונתון על ידי:

$$K \simeq \lceil \log(2((p+1) + \sum_{j=1}^{p+1} EC_j)) \rceil$$

שלב II (קידוד הסדרות)

זמן יצירת מילות קוד

זמן קידוד המחרוזת ה-j-ית נתון על ידי $O(\ell j)$ כאשר ℓj הוא אורך המחרוזת ה-j-ית. במימוש זה אין צורך בעידכון העץ מכיוון שהעץ קבוע.

הקצבת סיביות

מספר הסיביות $\hat{k}$ הדרוש לכל מילת קוד נתון על ידי:

$$\hat{k} \simeq \lceil \log((p+1) + \sum_{j=1}^{p+1} EC_j) \rceil \qquad (3.2)$$

כאשר p : מספר המחרוזות שהתקבלו בעת קידוד של הסדרה האופיינית בשלב I.

EC_j : מספר הסיביות הנוספות.

לכן מס' הסיביות לאות מקור לאחר הקידוד נתון על ידי הביטוי:

$$\rho_{UT}(u) = \frac{\hat{k}\,\hat{p}}{N} \qquad (3.3)$$

כאשר $\hat{p}$: מספר המחרוזות שנוצרות בעת הקידוד (שלב II) בעזרת עץ הפיסוק האוניברסלי, ו-N : מספר הסימבולים בסדרה.

פיענוח

פעולת המפענח במימוש זה מצטמצמת לחיפוש בטבלת התרגום שהיא עץ הפיסוק האוניברסלי שנבנה בשלב I של הקידוד. זמן החיפוש של מילות קוד בטבלת התירגום נתון על ידי $0(1)$. לא דרוש זמן לעידכון. כמות הזכרון הדרושה לאיחסון טבלת התרגום (עץ פיסוק אוניברסלי) נתונה על ידי ביטוי (3.1).

3.3.4 איחסון עץ אוניברסלי

אחד החסרונות של מימוש האלגוריתם על ידי עץ אוניברסלי נובע מהצורך לאחסן את מבנה הנתונים המייצג את העץ (למשל טבלה, רשימה מקושרת) בזכרון המחשב (דיסק או סרט), כך שבשעת הצורך ניתן יהיה לפענח את מילות הקוד שהתקבלו בעזרת העץ האוניברסלי.

כיוון שהזכרון הנדרש לאחסון מספר עצים עשוי להיות גדול, מוצגת כאן אפשרות אחרת והיא אחסון של מילות הקוד (באורך משתנה שנפלטות בעת בנית העץ תוך קידוד הסדרה האופיינית ולא העץ עצמו. דרך זו חוסכת זכרון אך מחייבת בניה מחודשת של העץ (טבלת התרגום) מתוך מילות הקוד הללו בכל פעם שיש להשתמש בו. כמות הזכרון הדרושה לאחסון מילות הקוד היא:

$$\hat{M}_R \simeq (p+1)\ \log(p+1)\ + \sum_{j=1}^{p+1} EC_j \qquad (3.4)$$

כאשר $\hat{M}_R < M_R$.

3.3.5 תוצאות מספריות

בטבלה הבאה מוצגים ערכי $\rho_{UT}(u)$ (מספר סיביות לפיקסל) המתקבלים לאחר הקידוד עבור שתי שיטות המימוש, קידוד ללא גרימת עוות (שיטה א') וקידוד תוך גרימת עוות (שיטה ב').

מכיוון שעבור אותה תמונה ניתן לקבל ערכים שונים של $\rho_{UT}(u)$ כאשר משתמשים בעצי פיסוק שונים נציג תוצאות עבור שני עצי פיסוק שונים. העץ ה-I בנוי מאותן 3 תמונות פנים המשמשות כקבוצת תמונות בקורת למימושים השונים.

העץ ה-II מורכב מחלקי תמונות פנים אחרות.

לצורך השוואה נציג את ערכי $\hat{H}(u)$ ו-$\rho(u)$ המתקבלים על ידי עץ פיסוק בינרי תוך שמירת גבולות ה-Byte.

טבלה מס' 3.3: ערכי $\rho_{UT}(u)$ המתקבלים עבור תמונות פנים בעזרת שני עצים אוניברסלים שונים, $\hat{H}(u)$ משערך לאנטרופיה ו-$\rho(u)$ מס' סיביות לפיקסל המתקבל על ידי המימוש ללא עץ אוניברסלי.

Table 3.3: $\rho_{UT}(u)$ values for three-pictures, coding done with two different universal trees. $\hat{H}(u)$: entropy estimator, $\rho(u)$: no. of bits/pixel, without universal tree.

תמונה	$\rho_{UT}(u)$				$\rho(u)$	$\hat{H}(u)$ (סיביות/פיקסל)
	עץ I		עץ II			
	ללא עוות	עם עוות	ללא עוות	עם עוות		
1	1.8922	1.6789	2.0416	1.7546	1.8206	1.3854
2	1.9581	1.7516	2.1810	1.9956	1.9890	1.5192
3	1.5527	1.4111	3.4202	3.2723	1.6444	1.2525
ממוצע	1.8010	1.6139	2.5476	2.3408	1.8180	1.3857

מתוך התוצאות שהוצגו מתקבלות המסקנות הבאות:

1) ביצועי מימוש UT-ZL רגישים למבנה עץ הפיסוק האוניברסלי. רגישות זו מתבטאת בהבדל המשמעותי בערכי $\rho_{UT}(u)$ המתקבלים על ידי קידוד בעזרת שני עצי הפיסוק, בתמונה מס' 3.

2) בשיטה ב' של מימוש UT-ZL (קידוד עם טעות) קיים שיפור של 0.2 סיביות לפיקסל מתוך 1.8 סיביות לפיקסל המתקבלות בשיטה ללא גרימת עוות. הבדל זה בערכי $\rho_{UT}(u)$ איננו מצדיק שימוש בשיטה ב'.

3) השוואת ערכי $\rho_{UT}(u)$ לערכי $\rho(u)$ אינה מוחלטת מפני שערכי $\rho_{UT}(u)$ משתנים בהתאם לעץ הפיסוק בו משתמשים לצורך הקידוד. ההבדל בין $\rho_{UT}(u)$ המתקבל מהעץ ה-I ל-$\rho(u)$ הוא קטן (0.02 סיבית מתוך 1.82) ואילו ההבדל בין אותם ערכים כאשר הקידוד מתבצע בעזרת עץ II הוא משמעותי (0.73 סיביות מתוך 1.82).

3.4 מימוש עץ אוניברסלי ב-4 מישורי סיבית נפרדים

מימוש זה שיסומן על ידי 4BP-UT-ZL הינו שילוב של שני המימושים שתוארו בפרק זה, כלומר עיקרון הקידוד מבוסס על שימוש בעצי פיסוק קבועים ונתונים מראש (מימוש UT) עבור כל אחד מ-4 מישורי הסיבית (מימוש 4BP).

אופן הקידוד: הקידוד נעשה על ידי המימוש UT על כל מישור סיבית לחוד לפי קוד GRAY של ערכי הפקסלים בעזרת עץ פיסוק בינרי רגיל. נציג שתי גירסאות של המימוש 4BP-UT-ZL. באחת נשתמש בעצי פיסוק אוניברסליים שונים המתאימים לכל אחד ממישורי הסיבית בנפרד, ובשניה נשתמש בעץ פיסוק אוניברסלי אחד ויחיד לכל מישורי הסיבית.

3.4.1 סכמת הקידוד

שלבי הקידוד זהים לאלה שבמימוש UT, אך במימוש 4BP-UT אין אפשרות לגירסה של קידוד עם טעות מפני שבתהליך הפיסוק משתמשים בעץ פיסוק בינרי רגיל על מנת לפסק את הסדרות הבינריות השייכות לכל מישורי הסיבית.
מימוש 4BP-UT-ZL פשוט יותר ממימוש UT-ZL כיוון שבראשון הקדוד הוא על סדרה בינרית ללא הגדרת גבולות ה-Byte.
במימוש 4BP-UT-ZL קיימת האפשרות לקידוד בזמן אמיתי (ON-LINE) או לקידוד ב-OFF-LINE כפי שקיים במימוש 4BP-ZL (ללא עץ אוניברסלי).

דיאגרמת הבלוקים של המערכת המבצעת את הקידוד לפי מימוש 4BP-UT זהה בשתי הגירסאות של מימוש 4BP-ZL פרט לכך שהבלוקים המממשים את הפיסוק הרגיל במימוש 4BP מוחלפים בבלוקים אשר מממשים את הפיסוק בעזרת עץ אוניברסלי במימוש 4BP-UT.

3.4.2 ניתוח המימוש

קידוד

שלב I: שלב זה זהה מכל הבחינות למימוש של האלגוריתם בעזרת עץ פיסוק בינרי רגיל. ראה ניתוח המימוש בסעיף 2.3.2.

כמות הזכרון הדרושה לאיחסון עץ אוניברסלי עבור כל מישור סיבית נתונה על ידי:

$$M_R \propto 2K(p_B+1) \qquad (3.5)$$

כאשר p_B: מספר המחרוזות שמתקבלות לאחר קידוד מישור סיבית אחד.

K : מספר הסיביות לייצוג כל אחד מהמחוונים המצביעים על צמתי העץ, כאשר: $K \simeq \lceil \log(2(p_B+1)) \rceil$.

שלב II: בשלב זה מקדדים את הסדרות השונות בעזרת העץ האוניברסלי הנתון; קיימות שתי אפשרויות:

א. שימוש בעצי פיסוק שונים המתאימים לכל מישור סיבית בנפרד.

ב. שימוש בעץ פיסוק אחד ויחיד לכל מישורי הסיבית.

יעילותן של 2 האפשרויות תקבע על פי ערכי $\rho_{4BP-UT}(u)$ בסעיף הבא. מבלי להתחשב בערכי $\rho_{4BP-UT}(u)$ אפשרות ב' עדיפה מהבחינות הבאות:

א. יש לאחסן עץ אוניברסלי אחד בלבד.

ב. התקורה הכרוכה בטיפול בעץ אחד נמוכה מזו הכרוכה בטיפול ב-4 עצים.

זמן יצירת מילות הקוד

זמן הקידוד של המחרוזת ה-j-ית נתון על ידי $O(\ell_j)$ כאשר ℓ_j: אורך המחרוזת ה-j-ית. לא דרוש זמן עידכון העץ.

הקצבת סיביות

מספר הסיביות $\hat{k}$ הדרוש לכל מילת קוד זהה ונתון על ידי:

$$\hat{k} = \lceil \log(p_B+1) \rceil \tag{3.6}$$

לכן יחס הדחיסה $\rho^i_{UT}(u)$ עבור מישור הסיבית ה-i-ית נתון על ידי:

$$\rho^i_{UT}(u) = \frac{\hat{k}^i\ \hat{p}^i_B}{N} \tag{3.7}$$

כאשר $\hat{p}^i_B$: מספר המחרוזות שנוצרות בעת הקידוד (שלב II) בעזרת עץ הפיסוק האוניברסלי,

N : מספר איברי הסדרה (מס' הסיביות במישור סיבית).

מספר הסיביות לאות מקור הוא סכום יחסי הדחיסה של כל מישורי הסיבית, דהיינו:

$$\rho_{UT}(u) = \frac{1}{N}\sum_{i=0}^{3} \hat{k}^i\ \hat{p}^i_B \tag{3.8}$$

פענוח

בדומה למימוש UT-ZL פעולת הפיענוח מצטמצמת לחיפוש בטבלת התירגום (עץ הפיסוק האוניברסלי), כלומר עם קבלת מילת קוד נפלטת המחרוזת המתאימה (סדרת סיביות).

זמן החיפוש בטבלה הוא O(1). לא דרוש זמן עידכון. כמות הזכרון הדרושה לאיחסון הטבלה (עץ הפיסוק) נקבעת על ידי הביטוי (3.5).

עבור המקרה בו מאחסנים את מילות הקוד (באורך משתנה) המיצגות את העץ האוניברסלי (ראה סעיף 3.3.4), כמות הזכרון הדרושה לאחסונו נתונה על ידי:

$$\hat{M}_R \simeq \lceil (p_B+1)\ \log(p_B+1) \rceil \tag{3.9}$$

כאשר $M_R < \hat{M}_R$.

3.4.3 תוצאות מספריות

בסעיף זה נציג את ערכי $\rho_{4BP-UT}(u)$ (מס' הסיביות לפיקסל) המתקבלים על ידי המימוש 4BP-UT-ZL ונשווה אותם עם מספרי הסיביות לפיקסל המתקבלים על ידי המימושים האחרים של האלגוריתם.

בטבלאות מס' 3.4 ו-3.5 מוצגים ערכי $\rho_{4BP-UT}(u)$ המתקבלים עבור 3 תמונות פנים. הקידוד התבצע בעזרת שני עצי פיסוק אוניברסלים שונים כפי שנעשה במימוש UT-ZL. עץ הפיסוק ה-I בנוי מחלקים של אותן 3 תמונות פנים שמשמשות כקבוצת תמונות ביקורת למימושים השונים. עץ הפיסוק ה-II מורכב מחלקי תמונות פנים אחרות. כל התמונות מיוצגות על פי קוד GRAY.

בטבלה מס' 3.4 מוצגים ערכי $\rho_{4BP-UT}(u)$ המתקבלים לאחר קידוד בעזרת 4 עצי פיסוק אוניברסליים המתאימים ל-4 מישורי סיבית. בטבלה מס' 3.5 מוצגים ערכי $\rho_{UT}(u)$ המתקבלים לאחר קידוד בעזרת עץ פיסוק יחיד לכל מישורי הסיבית. במקרה זה משתמשים בעץ פיסוק אוניברסלי השייך למישור הסיבית המשמעותית ביותר (MSB).

בטבלאות מוצגים ערכי משערך האנטרופיה $\hat{H}(u)$ ומספר הסיביות לפיקסל המתקבל על ידי המימוש 4BP-ZL (ללא עץ אוניברסלי) $\rho_{4BP}(u)$.

טבלה מס' 3.4: ערכי $\rho_{4BP-UT}(u)$ (מימוש 4BP-UT-ZL) המתקבלים על ידי שימוש בעצי פיסוק יחודיים עבור כל מישור סיבית.

Table 3.4: $\rho_{4BP-UP}(u)$ values achieved by using a different parsing tree for each bit plane.

(סיביות/פיקסל)	$\hat{H}(u)$	$\rho_{4BP}(u)$	$\rho_{4BP-UT}(u)$		תמונה
			עץ II	עץ I	
	1.3854	1.7337	1.7256	1.6797	1
	1.5192	1.6811	1.6344	1.6154	2
	1.2525	1.4348	1.4214	1.3548	3
	1.3857	1.6166	1.5938	1.5500	ממוצע

טבלה מס' 3.5: ערכי $\rho_{4BP-UT}(u)$ המתקבלים על ידי שימוש בעץ פיסוק אוניברסלי יחיד לכל מישורי הסיבית.

Table 3.5: $\rho_{4BP-UT}(u)$ values using one parsing tree for all bit planes.

$(\frac{\text{סיביות}}{\text{פיקסל}})$	$\hat{H}(u)$	$\rho_{4BP}(u)$	$\rho_{4BP-UT}(u)$		תמונה
			עץ II	עץ I	
	1.3854	1.7337	1.7396	1.7205	1
	1.5192	1.6811	1.6920	1.6700	2
	1.2525	1.4348	1.4427	1.4165	3
	1.3857	1.6166	1.6248	1.6023	ממוצע

התוצאות המוצגות בטבלאות מצביעות על הנקודות הבאות:

1) בשיטה בה משתמשים בעצי פיסוק יחודיים לכל מישור סיבית מתקבל שערכי $\rho_{4BP-UT}(u)$ נמוכים (בשני עצי הפיסוק) מערכי $\rho_{4BP}(u)$.

2) בשיטה בה משתמשים בעץ פיסוק אחד לכל מישורי הסיבית מתקבל שערכי $\rho_{UT}(u)$ נמוכים מערכי $\rho_{4BP}(u)$ עבור עץ הפיסוק ה-I. עבור עץ הפיסוק ה-II ערכי $\rho_{UT}(u)$ גבוהים מעט (בממוצע 0.0082 סיביות מתוך 1.6166) מערכי $\rho_{4BP}(u)$.

3) בין אם משתמשים בשיטה של עצי פיסוק יחודיים לכל מישור סיבית ובין אם משתמשים בעץ פיסוק אחד לכל מישורי הסיבית מתקבל שאין הבדל משמעותי בין עץ I לעץ II במספר הסיביות לפיקסל המתקבל לאחר הקידוד. עובדה זו מצביעה על כך שמימוש 4BP-UT-ZL אינו רגיש למבנה עץ הפיסוק.

3.4.4 השוואה בין המימושים UT-ZL ו-4BP-UT-ZL

בטבלה הבאה נשווה את ערכי $\rho_{UT}(u)$ של המימושים UT-ZL ו-4BP-UT-ZL. ערכי $\rho(u)$ של המימוש 4BP-UT מתקבלים על ידי עצי פיסוק יחודיים לכל מישורי הסיבית.

טבלה מס' 3.6: ערכי $\rho_{UT}(u)$ ו-$\rho_{4BP-UT}(u)$ המתקבלים על ידי 2 עצים אוניברסליים שונים.

Table 3.6: $\rho_{UT}(u)$ and $\rho_{4BP-UT}(u)$ values received through two different universal trees.

סיביות/פיקסל	מימוש UT-ZL		מימוש 4BP-UT-ZL		תמונה
	עץ II	עץ I	עץ II	עץ I	
	2.0416	1.8922	1.7396	1.6797	1
	1.9956	1.9581	1.6920	1.6154	2
	3.2733	1.5527	1.4427	1.3548	3
	2.5476	1.8010	1.5938	1.5500	ממוצע

מהתוצאות שבטבלה ניתן לראות:

1) ערכי $\rho_{4BP-UT}(u)$ (מימוש 4BP-UT) נמוכים מערכי $\rho_{UT}(u)$ (מימוש UT-ZL).

2) כאשר משתמשים בעץ פיסוק אוניברסלי מספר II הערך הממוצע של $\rho_{4BP-UT}(u)$ נמוך ב-0.96 סיבית ביחס לערך הממוצע של $\rho_{UT}(u)$ שהוא 2.55 סיבית לפיקסל. זהו הפרש גדול המורגש בעיקר בתמונה מס' 3 בה מספר הסיביות לפיקסל הוא 3.27 לעומת 1.44 המתקבלות על ידי מימוש 4BP-UT-ZL.

סיכום

יתרונות המימוש המשולב (4BP-UT-ZL) ביחס למימושים אחרים:

א. אורך קבוע של מילות קוד;

ב. חסינות טובה יותר לרעש;

ג. סיבוכיות נמוכה של תהליך הקידוד והפיענוח;

ד. יכולת דחיסה טובה (ערכי $\rho(u)$ נמוכים);

ה. אפשרות לשימוש בעץ אוניברסלי יחיד לכל מישורי הסיבית;

ו. רגישות נמוכה של ביצועיו למבנה הסטטיסטי של עץ הפיסוק האוניברסלי.

חסרונות המימוש:

א. טיפול נפרד בכל מישור סיבית.

ב. יצירת מילות קוד בקצב לא קבוע לכל מישור.

ג. אי התאמה בין אורך הסדרות המקודדות במישורים השונים.

על סמך יתרונות אלה מימוש זה נראה עדיף על פני המימושים האחרים שנבחנו.

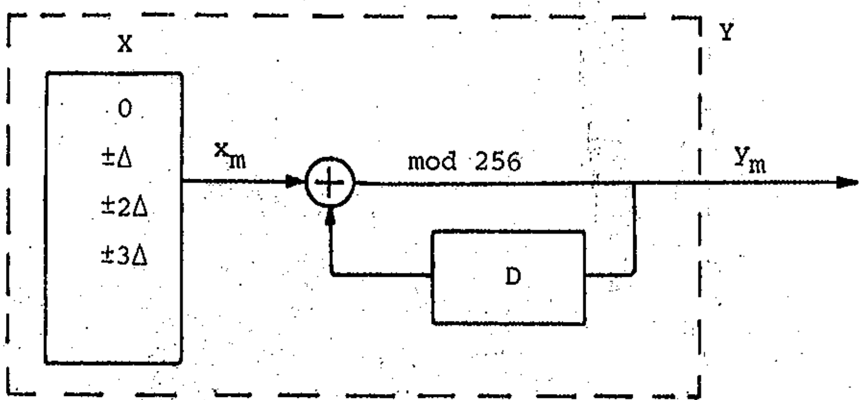
3.5 ניתוח השוואתי של המימושים UT-ZL ו-4BP-UT-ZL

המניע לניתוח ההשוואתי של המימושים UT-ZL ו-4BP-UT-ZL נעוץ בהבדל המשמעותי שקיים בין רגישויות יכולת הדחיסה של כל אחד מהם בתלות בעץ הפיסוק האוניברסלי שבעזרתו מתבצע הפיסוק. נבדוק את השתנות ערכי הדחיסה $\rho_{4BP-UT}(u)$ ו-$\rho_{UT}(u)$ וננתח את הסיבות האפשריות לכך.

3.5.1 מודל יצירת תמונה

נעזר במודל ליצירת תמונה שתאורו המלא מופיע בפרק 5. בעזרת המודל נבנה תמונות סינטטיות (מלאכותיות) תוך שליטה מלאה על הפרמטרים המאפיינים אותן. באמצעות תמונות אלה ננתח ונבדוק את רגישות יכולת הדחיסה של 2 המימושים.

סכמת המודל:



ציור 3.5: מודל יצירת תמונה.

Fig. 3.5: Picture production model.

כאשר: $y_m = (y_{m-1} + x_m) \bmod 256$.

המקור X הינו מקור ללא זכרון אשר פולט את הסימבולים $0, \pm\Delta, \pm 2\Delta, \pm 3\Delta$.

Δ הוא גודל צעד הקואנטיזציה של המקור Y. עבור תמונות המיוצגות על ידי 4 סיביות $\Delta = 256/2^4$. הסתברות הופעת הסימבולים של המקור X מסומנת בצורה הבאה:

$$i = 1,2,3, Pr\{x_m=\pm i\Delta\} = P_i, Pr\{x_m=0\} = P_o$$

מודל יצירת התמונה מיוצג על ידי המקור Y, שהוא מקור מרקובי מסדר ראשון, אשר פולט מ"א y_m המהווים תהליך אקראי סטציונרי. באמצעות תהליך זה מנסים לתאר תמונות אמיתיות אף על פי שבדרך כלל אינן סטציונריות.

לאחר הצגת המודל יש לבדוק את התאמתו לתמונות פנים. הבדיקה בוצעה ותוצאותיה מוצגות בפרק 4.

ניתן לראות שלהסתברות הופעת הסימבול "0" של המקור "X" יש השפעה רבה על אנטרופית "התמונה" המתקבלת מהמקור Y, משום שהערך P_o קובע את "דרגת הפעילות" (תכיפות השינויים) בתמונה. ככל ש-P_o גבוה יותר (קרוב ל-1) הפעילות בתמונה פוחתת (מתקבלים מ"א y_m עוקבים בעלי אותו ערך בהסתברות גבוהה יותר) ולכן האנטרופיה יורדת.

נציג בטבלה הבאה את ערכי P_o, הסתברות הופעת הסימבול "0" של המקור X השייכים לתמונות פנים אמיתיות. יש לציין שערכי P_o הם מקורבים בלבד בגלל אי הסטציונריות של התמונות.

טבלה מס' 3.7: ערכי P_o של תמונות פנים שונות.

Table 3.7: P_o values for face pictures.

תמונה	1	2	3	4	5	6
P_o	0.71	0.66	0.72	0.65	0.59	0.66

תמונות הפנים 1, 2 ו-3 מהוות קבוצת תמונות ביקורת לבדיקת ביצועי (דחיסה) המימושים השונים ובפרט המימושים UT-ZL ו-4BP-UT-ZL. במימושים הנ"ל הקידוד בוצע בעזרת שני עצי פיסוק אוניברסליים, העץ ה-I נבנה בעזרת תמונות 1, 2 ו-3 והעץ ה-II נבנה בעזרת תמונות 4, 5 ו-6.

כדי לבצע בדיקה השוואתית על פרמטר אחד ויחיד ינתן לכל עץ פיסוק ערך P_o שיקבע על פי התמונות שבעזרתן העץ נבנה;

עבור העץ ה-I $\quad \bar{P}_o = 0.70$,

עבור העץ ה-II $\quad \bar{P}_o = 0.64$,

כאשר $\bar{P}_o$ מסמן ערך ממוצע של P_o.

למרות שההבדל בין ערכי $\bar{P}_o$ של שני העצים איננו גדול מס' הסיביות לפיקסל, המתקבל לאחר הקידוד במימוש UT-ZL, שונה במידה משמעותית עבור שני העצים, ואילו במימוש 4BP-UT-ZL אין הבדל משמעותי (ראה טבלה מס' 3.6 בעמ' 51).

3.5.2 תמונות סינטטיות: מדידת אנטרופיה

באמצעות המודל ליצירת תמונה נבנו מספר תמונות שבעזרתן נבדוק את התכונות השונות של מימוש האלגוריתם. גודל התמונות הוא 256×256 פיקסל. נסווג את התמונות ל-3 קבוצות על פי ערכי P_o המאפיינים אותן.

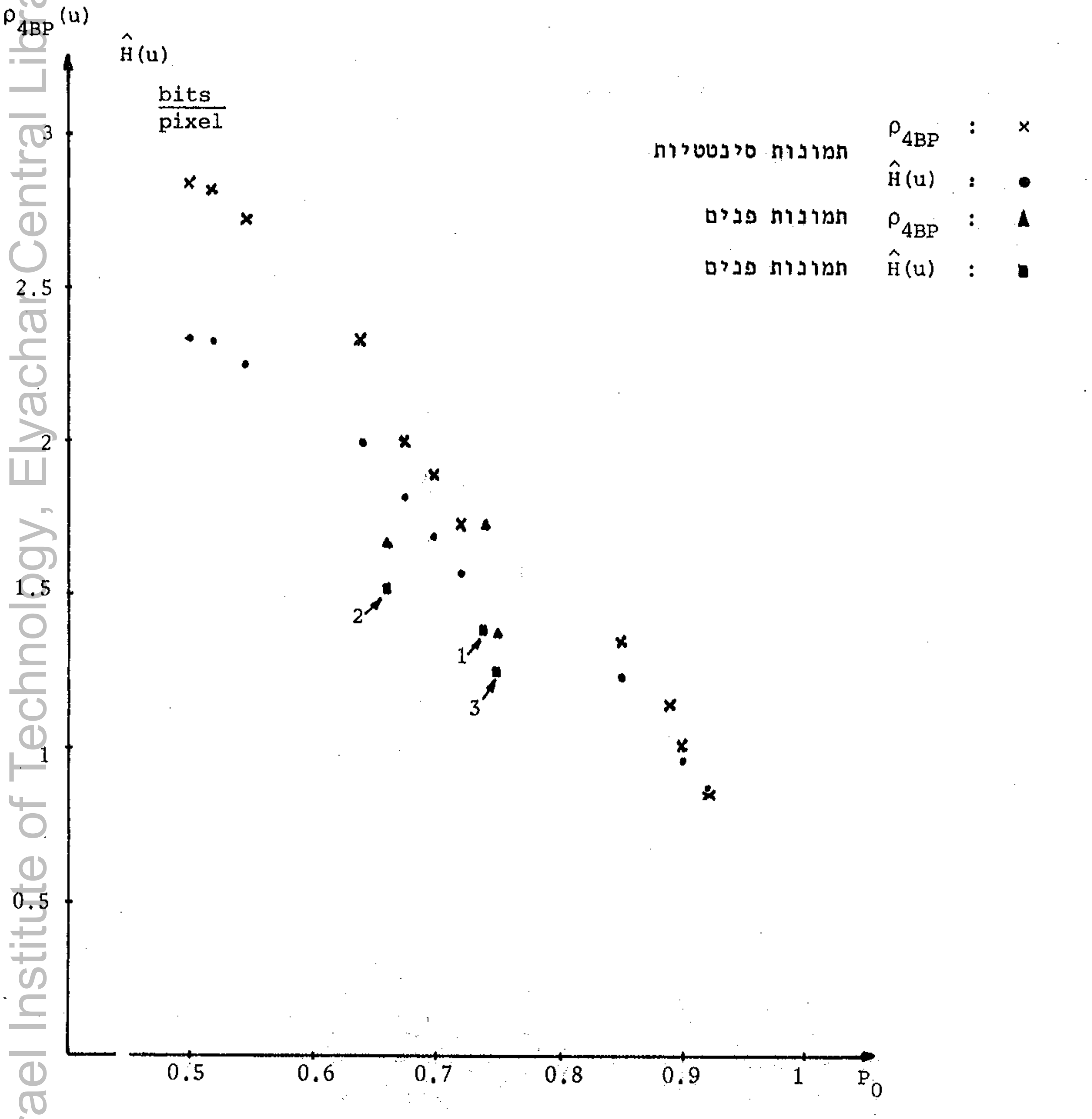
הקבוצות הן:

A: תמונות בעלות P_o בסביבה של 0.9,

B: תמונות בעלות P_o בסביבה של 0.7,

C: תמונות בעלות P_o בסביבה של 0.5.

בגרף הבא ניתן לראות את השתנות ערכי $\hat{H}(u)$ ו-$\rho_{4BP}(u)$ כפונקציה של P_o המתקבלים מתמונות סינטטיות המיוצגות על ידי 4 סיביות. בגרף מוצגים גם ערכי $\hat{H}(u)$ ו-$\rho_{4BP}(u)$ השייכים ל-3 תמונות הפנים. יש לזכור שהערך $\hat{H}(u)$ מתקבל על ידי המימוש המבוסס על עץ פיסוק בינרי תוך שמירת גבולות ה-Byte.



גרף מס' 3.1: ערכי $\hat{H}(u)$ ו-$\rho_{4BP}(u)$ של תמונות סינטטליות ושל תמונות פנים.

Graph 3.1: $\hat{H}(u)$ and $\rho_{4BP}(u)$ of syntetic and face pictures.

מנתוני הגרף עולות הנקודות הבאות:

א. כצפוי ערכי $\hat{H}(u)$ ו-$\rho_{4BP}(u)$ נמוכים יותר ככל ש-P_O קרוב יותר ל-1.

ב. ההבדל בערכי $\hat{H}(u)$ ו-$\rho_{4BP}(u)$ בין תמונות פנים לבין תמונות סינטטיות מצביע על כך שהמודל לא מייצג בשלמות, אם כי בצורה סבירה, את התכונות הסטטיסטיות של התמונות.

ג. הפער בין ערכי $\hat{H}(u)$ לבין ערכי $\rho_{4BP}(u)$ קטן ככל ש-P_o גבוה יותר.
סיבה אפשרית לכך שהתכנסות במימוש 4BP-ZL מהירה יותר מאשר במימוש המבוסס על עץ פיסוק בינרי תוך שמירת גבולות ה-Byte היא שבמימוש 4BP מפסקים סדרות בינריות בעלות רצפים ארוכים של אותו סימבול (0 או 1) ולא סדרות בעלות איברים השייכים לאלפבית מורכב יותר.

ראוי לציין כי נמצא בבדיקה שההבדל בין ערכי $\hat{H}(u)$ לבין ערכי $H(y_n/y_{n-1})$ של תמונות סינטטיות הוא מזערי (בממוצע 0.0003 סיביות מתוך 1.5).
אי הזהות נובעת כנראה מהעובדה שהתמונות הנבדקות הן סופיות.

3.5.3 מדד לביצועי המימושים

בדיקת השפעת מבנה העץ על יכולת הדחיסה של המימושים לעיל מבוססת על השוואת ערכי ρ_{UT} ו-ρ_{4BP-UT}. השוואה כזו נוחה יותר אם נגדיר מדד מתאים.
נגדיר את המדדים הבאים:

$$CR_H = \frac{\hat{H}(u)}{\rho_{UT}(u)} \tag{3.10}$$

$$CR_{BP} \triangleq \frac{\rho_{4BP}}{\rho_{4BP-UT}} \tag{3.11}$$

המדדים CR_H ו-CR_{BP} מבטאים את היחס בין ערכי הדחיסה המתקבלים במימושים המבוססים על עץ פיסוק אוניברסלי (UT-ZL ו-4BP-UT-ZL) לבין ערכי הדחיסה המתקבלים במימושים ללא עץ אוניברסלי (פיסוק בינרי תוך שמירת גבולות ה-Byte ו-4BP בהתאמה). לדוגמא, כאשר $CR_{BP} = 1$ אין שינוי בערכי הדחיסה (סיביות לפיקסל) המתקבלים על ידי שני המימושים: 4BP (ללא עץ אוניברסלי) ו-4BP-UT המבוסס על עץ פיסוק אוניברסלי. כאשר הגודל $CR_H = 0.5$, מספר הסיביות לפיקסל המתקבל על ידי המימוש המבוסס על עץ פיסוק אוניברסלי (UT-ZL) גדול פי 2 מזה המתקבל על ידי המימוש ללא עץ אוניברסלי.

אף על פי ששני הגדלים CR_H ו- CR_{BP} מודדים את יחס הדחיסה קיים הבדל קטן בהגדרתם. $\hat{H}(u)$ הוא חסם תחתון למספר הסיביות לפיקסל שאינו מכיל את מחיר הקידוד ואילו $\rho_{4BP}(u)$ הוא מספר הסיביות לפיקסל הכולל את מחיר הקידוד.

3.5.4 בדיקת רגישות המימושים

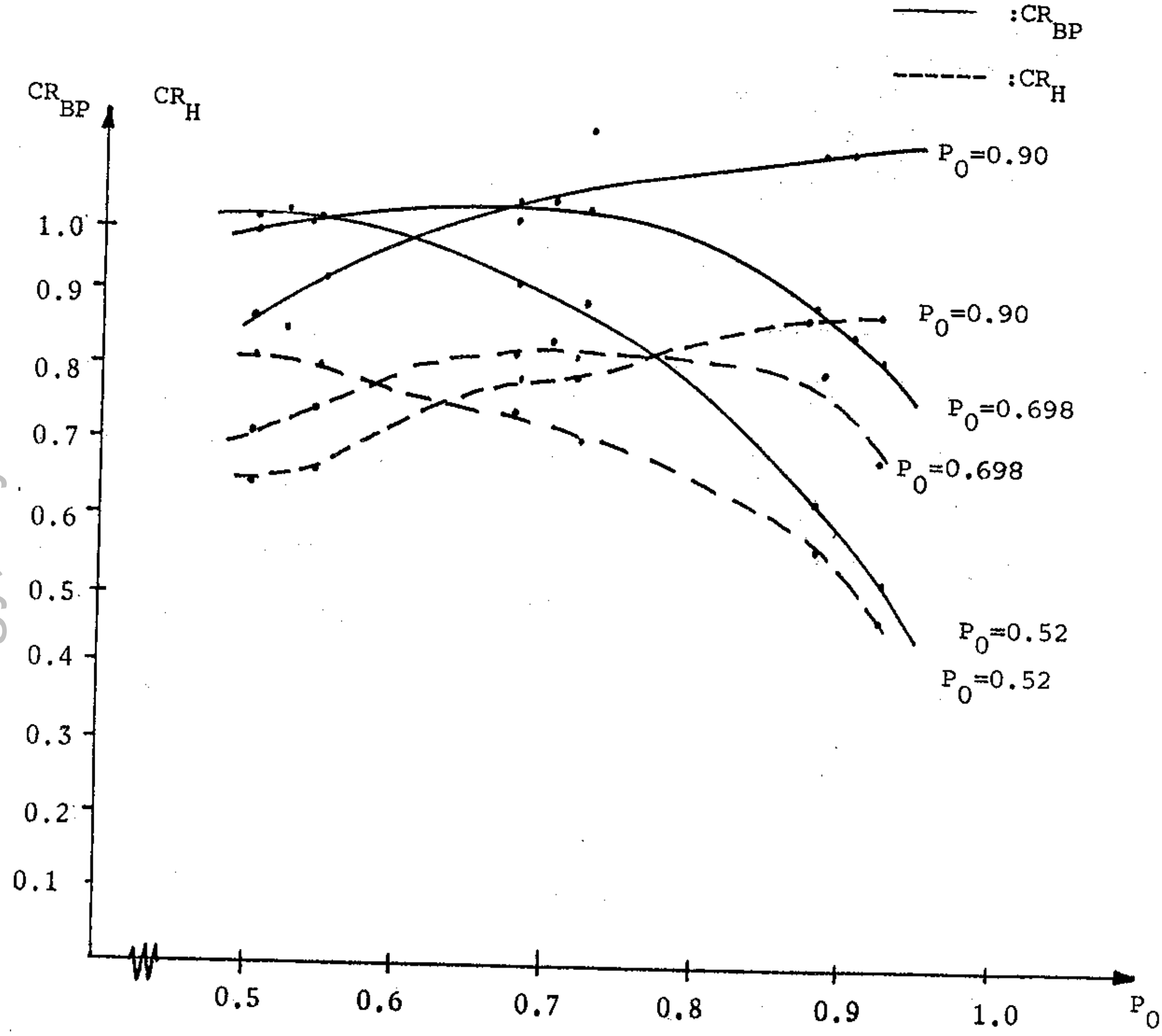
בדיקת רגישות יכולת הדחיסה של המימושים UT-ZL ו-4BP-UT-ZL תעשה במספר שלבים, שבמהלכם נבדוק את השתנות ערכי CR_H ו- CR_{BP} כפונקציה של המבנה הסטטיסטי של עץ הפיסוק ושל מודל יצירת התמונה.

שלב I: עצי פיסוק סטציונריים

עץ פיסוק נקרא סטציונרי כאשר התמונה או התמונות מהן העץ נבנה הן סטציונריות, דהיינו ערכי P_O נשארים קבועים.
בעזרת תמונות סינטטיות יצרנו 3 עצי פיסוק סטציונריים שונים בעלי P_O המתאימים לכל אחת מ-3 קבוצות התמונות הסינטטיות. ערכי P_O של העצים הם:
עץ A: $P_O = 0.90$; עץ B: $P_O = 0.698$; עץ C: $P_O = 0.52$.

בגרף מס' 3.2 מוצגים ערכי CR_H ו- CR_{BP} המתקבלים עבור העצים A, B ו-C כפונקציה של P_O של התמונות השונות.

מהגרף מתקבל שעבור כל אחד מעצי הפיסוק ההשתנות של ערכי CR_H ו- CR_{BP} כפונקציה של P_O היא דומה. לעומת זאת, עבור תמונות אמיתיות ערכי CR_{BP} נשארים כמעט קבועים ללא תלות בעץ הפיסוק וערכי CR_H משתנים באופן ניכר כפונקציה של עץ הפיסוק. למרות האמור לעיל התנהגות העקומות בגרף מצביעה על הנקודות הבאות:



גרף מס' 3.2: ערכי CR_H ו-CR_{BP} המתקבלים עבור 3 עצי פיסוק שונים.

Fig. 3.2; CR_H and CR_{BP} values for 3 parsing trees.

א. עבור תמונות סינטטיות בהן P_o קרוב לזה של עץ הקידוד מתקבל שערכי CR_{BP} שווים או גדולים מ-1. בתמונות השייכות לקבוצה A ($P_o \approx 0.9$) העוברות קידוד על ידי העץ A ($P_o = 0.9$) ערכי CR_{BP} הם הגבוהים ביותר (1.12). השיפור בדחיסה המתקבל על ידי המימוש 4BP-UT במקרים הנ"ל נובע מכך שהסדרות האופייניות של המקורות בעלי P_o מסויים כבר נמצאות "במילון" שהוא העץ האוניברסלי. לכן, הקידוד יעיל יותר מכיוון שאין צורך בתהליך הלמידה של התכונות הסטטיסטיות של התמונות כפי שנעשה במימוש הרגיל. כאשר התמונות הן

בעלות אנטרופיה נמוכה $(P_o \simeq 0.9)$ התכנסות אלגוריתם ZL במימוש ללא עץ אוניברסלי איטית משום שתהליך הלמידה של התכונות הסטטיסטיות ארוך ולכן שיפור בדחיסה המושגת על ידי המימוש המבוסס על עץ אוניברסלי בולט יותר.

ב. ערכי CR_H ו-CR_{BP} נמוכים יותר כאשר מקדדים תמונות השייכות לקבוצה A $(P_o \simeq 0.90)$ בעזרת העץ C $(P_o = 0.52)$ מאשר במקרה הפוך, דהיינו קידוד של תמונות השייכות לקבוצה C $(P_o \simeq 0.50)$ בעזרת העץ A $(P_o = 0.90)$.
הסיבה לכך נעוצה במבנה של עצי הפיסוק. בעץ בו $P_o = 0.9$ קיימים ענפים ארוכים (סדרות סימבולים ארוכים) לעומת זאת, עץ בו $P_o = 0.52$ הינו סבוך וענפיו קצרים. מכאן שקידוד תמונות בהן $P_o \simeq 0.9$ בעזרת עץ בו $P = 0.52$ פוגם ביכולת הדחיסה משום שלעץ חסרים אותם ענפים ארוכים הדרושים לקידודו היעיל של התמונות. לעומת זאת הפגיעה בביצועי האלגוריתם במקרה ההפוך פחותה משום שבתהליך הקידוד של תמונות בהן $P_o \simeq 0.50$ לא נחוצים ענפים ארוכים והעץ בו $P_o = 0.9$ הינו סבוך מספיק מכיוון שהוא מכיל ענפים קצרים בעת התפתחות ענפיו הארוכים.

ג. ערכי CR_H ו-CR_{BP} המתקבלים בעזרת העץ B $P_o = 0.7$ הם היציבים ביותר מבין אלה המתקבלים מכל העצים; תכונה זאת רצויה לניהול חוצץ בערוצי תקשורת הדרוש לקבלת קצב קבוע של שידור מילות קוד.

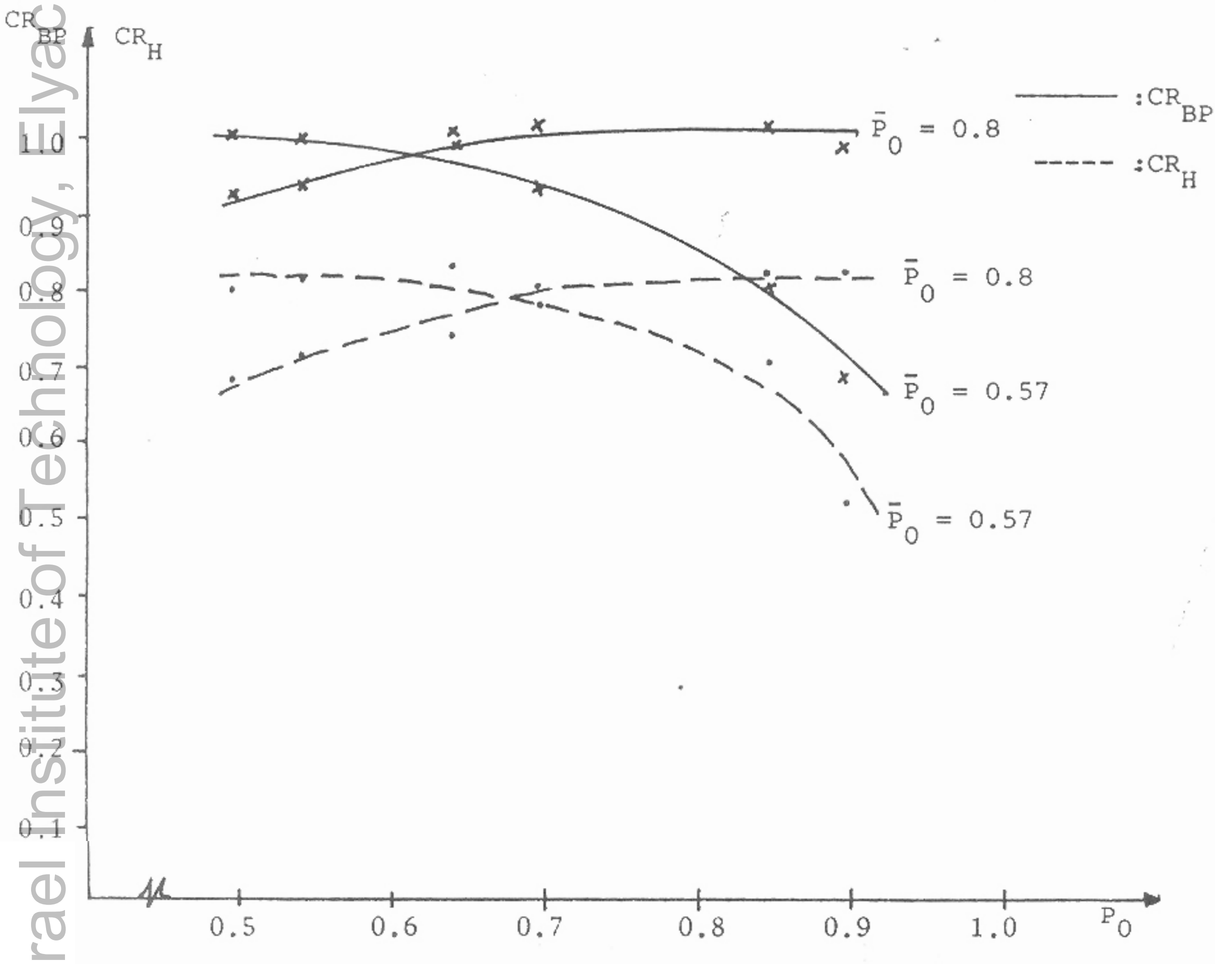
<u>שלב II: עצי פיסוק לא סטציונריים</u>

בבדיקה שבוצעה בשלב הקודם לא התקבלה התופעה שבתמונות אמיתיות למימושים UT-ZL ו-4BP-UT-ZL יש רגישות שונה למבנה העץ. לכן, בשלב זה נבדוק את השפעתו של עץ פיסוק לא סטציונרי על רגישות המימושים, כיוון שתמונות פנים אינן סטציונריות. עץ פיסוק נקרא לא סטציונרי כאשר התמונה או התמונות מהן העץ נבנה הן בעלות ערכי P_o לא קבועים.
לבדיקת השתנות ערכי CR_H ו-CR_{BP} יצרנו 2 עצי פיסוק לא סטציונריים כאשר בכל אחד מהם ערכי P_o של התמונות שבעזרתן נבנו העצים קרובים אך שונים (טווח של ± 0.07 עד ± 0.1 בסביבת $\bar{P}_o$).

עבור עץ D $\bar{P}_o = 0.80$.

עבור עץ E $\bar{P}_o = 0.57$.

בגרף הבא מוצגים ערכי CR_H ו-CR_{BP} המתקבלים עבור העצים הלא סטציונריים D ו-E כפונקציה של P_o של התמונות הסינטטיות השונות.



<u>גרף מס' 3.3</u>: ערכי CR_H ו-CR_{BP} המתקבלים עבור שני עצי פיסוק לא סטציונריים.

<u>Graph 3.3</u>: CR_H and CR_{BP} for 2 non-stationary parsing trees.

שוב מתקבל שעבור כל אחד מעצי הפיסוק השתנות של ערכי CR_H ו-CR_{BP} כפונקציה של P_o דומה, ויתר על כן התנהגות העקומות הנ"ל דומה לזו של אלה שהתקבלו עבור עצי פיסוק סטציונריים.

שלב III: מודל נוסף ליצירת תמונה

הבנת תהליכי הקידוד לא שופרה על ידי הבדיקות שבוצעו לעיל בעזרת שני סוגים של עצי פיסוק, עץ סטציונרי ועץ לא סטציונרי. לשם כך נבדוק בשלב זה מודל אחר ליצירת תמונה.

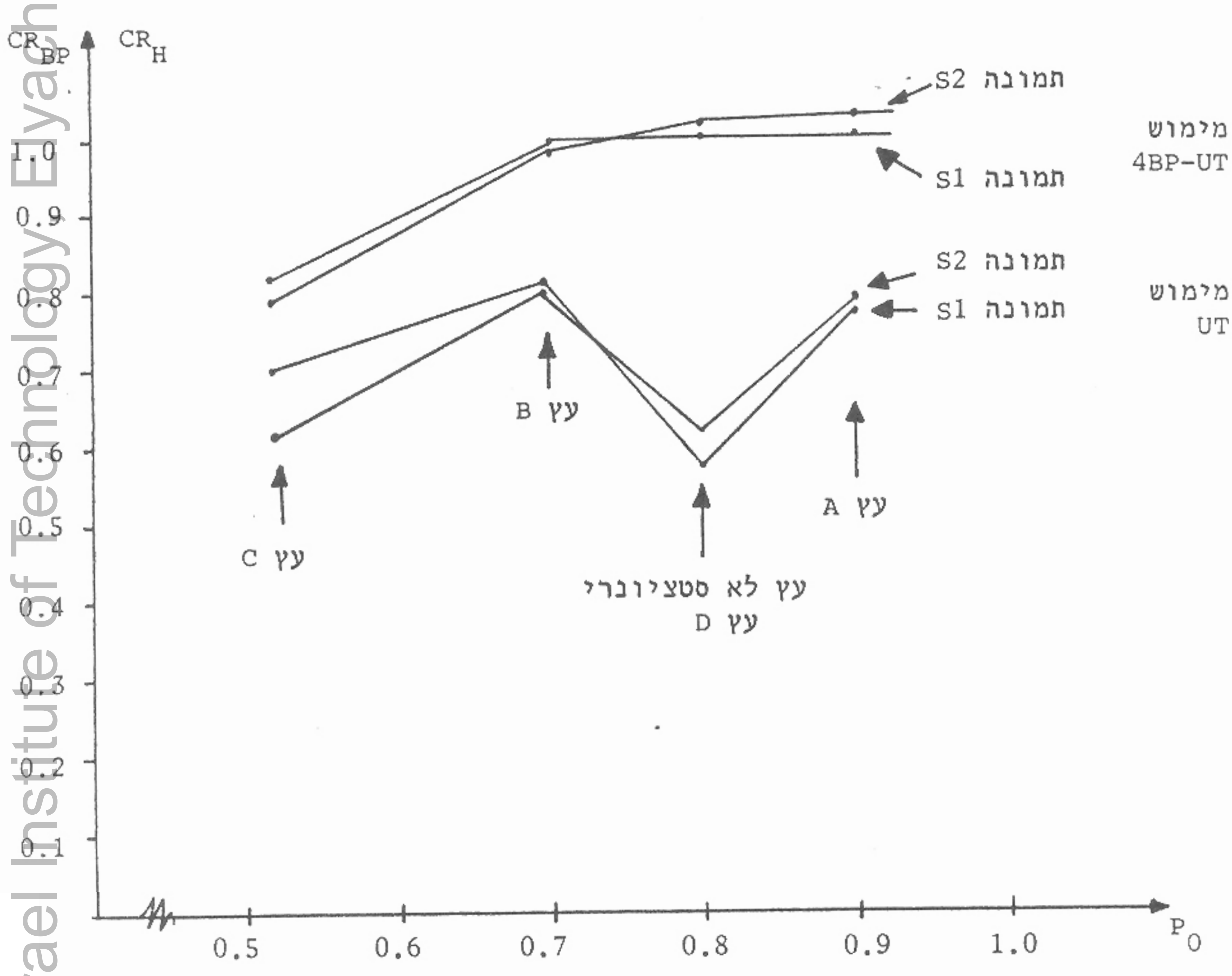
המודל החדש זהה במבנהו למודל יצירת התמונה שתואר קודם, פרט לכך שהמקור חסר הזכרון X הוחלף במקור מרקובי מסדר ראשון X. כלומר, המודל החדש מבוסס על שירשור של שני מקורות מרקוביים מסדר ראשון כמתואר בסוף פרק 4.

בחירה מתאימה של ערכי ההסתברות המותנית של סימבולי המקור X מאפשרת קבלת רצפים ארוכים של סימבולים בעלי אותו ערך ביציאת המקור Y. רצפים ארוכים אלה קיימים בתמונות בעלות אופי Low Pass או בתמונות בהן מספר הפרטים מועט, דוגמא לכך היא תמונה פנים מס' 3. יש לזכור שבקידוד תמונה זו נמצא הבדל משמעותי בין ערכי הדחיסה המתקבלים על ידי המימוש UT-ZL לבין אלה המתקבלים על ידי המימוש ללא עץ אוניברסלי.

כדי לערוך את הבדיקה יצרנו 2 תמונות סינטטיות s1 ו-s2 בעזרת המודל החדש (שירשור 2 מקורות מרקוביים מסדר ראשון). מכיוון שהשליטה על תכונות המקור $\hat{X}$ נעשית על ידי קביעת ערכי ההסתברות המותנית של הופעת הסימבולים השונים, הערך P_o במקרה זה המתאים ל-2 התמונות מתקבל על ידי מדידה על התמונות עצמן. הערך P_o עבור תמונה s1 הוא $P_o = 0.686$ ועבור תמונה s2 הוא $P_o = 0.736$.

ערכי P_o של תמונות s1 ו-s2 קרובים לאלה של קבוצת תמונות הביקורת. ראה טבלה 7 בעמוד 53.

בגרף הבא מופיעים ערכי CR_H ו-CR_{BP} המתקבלים מהתמונות S1 ו-S2 כפונקציה של ערכי P_o של עצי הפיסוק האוניברסליים. עצי הפיסוק הסטציונריים הם: עץ A: $P_o = 0.9$, עץ B: $P_o = 0.698$ ועץ C: $P_o = 0.52$. עץ הפיסוק הלא סטציונרי הוא עץ D בו $\bar{P}_o = 0.8$.



גרף מס' 3.4: ערכי CR_H ו-CR_{BP} המתקבלים עבור שתי תמונות סינטטיות כפונקציה של P_o של עצי הפיסוק האוניברסליים.

Graph 3.4: CR_H and CR_{BP} of two syntetic pictures vs. P_o values of the parsing trees.

על סמך תוצאות הבדיקה ניתן לקבוע שהמימוש UT-ZL רגיש למודל יצירת תמונה כאשר עץ הפיסוק איננו סטציונרי, זאת בניגוד למימוש 4BP-UT-ZL שאינו מושפע ממודל יצירת התמונה. התנהגות ערכי CR_H ו-CR_{BP} עבור תמונות פנים אמיתיות(המוצגים בטבלה הבאה) דומה לזו שהתקבלה בבדיקה זו.

לפני שנסיק מסקנות לגבי רגישות יכולת הדחיסה של שני המימושים נבדוק האם היצוג לפי קוד GRAY הוא הגורם לכך שהמימוש 4BP-UT חסין למבנה הסטטיסטי של התמונה ושל עץ הפיסוק. התשובה לשאלה נמצאת בטבלה הבאה בה מוצגים ערכי CR_{BP} של 3 תמונות פנים שעברו קידוד על ידי עץ הפיסוק ה-II (ראה סעיף 3.3.5). לצורך ההשוואה מוצגים גם ערכי CR_H.

טבלה מס' 3.8: ערכי CR_{BP} של תמונות פנים ביצוגן הבינרי ולפי קוד GRAY, ערכי CR_H של אותן התמונות.

Table 3.8: CR_{BP} and CR_H values of tree pictures for binary and GRAY code representation.

התמונה	CR_{BP}		CR_H
	יצוג GRAY	יצוג בינרי	ללא תלות ביצוג
1	0.94	1.00	0.68
2	1.09	1.03	0.70
3	0.99	1.01	0.44

מכאן, שהחלוקה ל-4 מישורי סיבית, ללא תלות ביצוג התמונה, היא הגורמת לכך שהמימוש 4BP-UT-ZL איננו רגיש למודל יצירת התמונה ולמבנה הסטטיסטי של עץ הפיסוק האוניברסלי.

השאלה הבאה שנשאלת היא: מהן הסיבות לכך שמימוש UT-ZL רגיש למבנה הסטטיסטי של עץ הפיסוק של מודל יצירת התמונה בניגוד לחוסר הרגישות של המימוש 4BP-UT-ZL?

תשובה לשאלה זאת דורשת בדיקות נוספות וניתוח אנליטי של תכונות העצים ושל המודל ליצירת תמונה שאותן לא נבצע במסגרת עבודה זו. לכן התשובה תנתן באופן חלקי וכהשערה.

סיבות אפשריות להבדל ברגישות של שני המימושים:

1) חלוקה ל-4 סיביות מטשטשת את התכונות הסטטיסטיות של המקור מכיוון שעל ידי החלוקה הקידוד נעשה על סדרה בינרית אשר מייצגת רק באופן חלקי את תכונות המקור.

2) קצב ההתכנסות של המימוש 4BP-ZL מהיר יותר מהסיבות הבאות:

א. עץ הפיסוק הותאם במיוחד לא"ב שלו (בינרי) ולא כפי שמתבצע במימוש השני בו מאלצים עץ בינרי לתפקד כעץ בעל דרגה α שהיא עוצמת הא"ב המקורי.

ב. בסדרות בעלות אורך סופי מספר המחרוזות האפשריות השונות קטן יותר בסדרות בינריות מאשר בסדרות לא בינריות (א"ב בעל $\alpha > 2$). לכן, עבור סדרת פיקסלים המתארת תמונה אחת עץ הפיסוק הבינרי הפועל על מישור סיבית אחד רוכש מהר יותר את התכונות הסטטיסטיות של המקור מאשר עץ פיסוק מדרגת α או עץ פיסוק בינרי תוך שמירת גבולות ה-Byte. זאת מפני שעבור אותו מספר סימבולים המחרוזות הבינריות שנוצרות בעת הפיסוק מייצגות אחוז גדול יותר מכלל המחרוזות האפשריות מאשר המחרוזות המתקבלות מעץ פיסוק לפי הא"ב האמיתי של התמונה. מהאמור לעיל ניתן להסיק שעץ שנבנה על ידי המימוש UT-ZL (א"ב מקורי של התמונה) מכיל פחות מידע על תכונות המקור מאשר עץ שנבנה על ידי המימוש 4BP-UT-ZL (א"ב בינרי) עבור מישור סיבית אחד בפרט כאשר העץ לא סטציונרי (כאשר מדובר בעץ אוניברסלי המיועד לקידוד תמונות בעזרת אותה שיטת פיסוק שהוא נבנה).

3.6 קידוד תמונות פנים בעזרת עץ סינטטי

אחד החסרונות של המימושים המבוססים על עץ פיסוק אוניברסלי נובע מהצורך להעביר למקלט (מפענח) את עצי הפיסוק שנוצרו בשלב הקידוד, או מהצורך לאחסן את עצי הפיסוק בזכרון המחשב.

דרך אפשרית להתגבר על חסרון זה היא ליצור עץ פיסוק אוניברסלי זהה לזה שבעזרתו התמונות קודדו, אם וכאשר יש צורך לפענח את מילות הקוד על מנת לקבל את התמונה מקורית מחדש. יצירת עץ חדש, זהה לזה שהיה בעת קידוד של

התמונות, אפשרית רק בתנאי שהעץ נבנה בעזרת תמונות סינטטיות מכיוון שרק אותן ניתן לשחזר במדויק על פי הפרמטרים של המודל. לבנית תמונה סינטטית דרושים מעט פרמטרים, כגון ערכי P_i, $i = 0,1,2,3$ של סימבולי המקור X וערך ה"גרעין" הדרוש לתוכנית ליצירת מספרים פסאודו אקראיים.

שיטה זו המבוססת על עץ פיסוק סינטטי אוניברסלי, יעילה רק בתנאי שערכי CR_H וערכי CR_{BP} קרובים לאלה המתקבלים עבור עצים של תמונות אמיתיות. לכן, נבדוק את האפשרות לקדד תמונות פנים על ידי עצים שנבנו בעזרת תמונות סינטטיות לפי מודל מרקובי מסדר ראשון. על מנת לקצר את תהליך הבדיקה נציג את התוצאות שמתקבלות מקידוד תמונה מס' 3 המאופיינת על ידי המודל המבוסס על שירשור של 2 מקורות מרקוביים מסדר ראשון. (ערכי CR_H שהתקבלו בצרוף של עץ פיסוק ותמונה השייכים למודלים שונים הם הנמוכים ביותר).

בטבלה הבאה מוצגים ערכי CR_H ו-CR_{BP} המתקבלים על ידי קידוד תמונה מס' 3 בעזרת עצים שונים.

טבלה מס' 3.9: ערכי CR_H ו-CR_{BP} המתקבלים על ידי עצים שונים עבור תמונה מס' 3.

Table 3.9: CR_H and CR_{BP} values received with different parsing trees for picture No. 3.

עץ סינטטי סטציונרי $P_o=0.70$	עץ D-סינטטי לא סטציונרי $\bar{P}_o=0.80$	עץ II תמונות פנים $\bar{P}_o = 0.63$	
0.65	0.52	0.44	CR_H
0.90	0.94	0.88	CR_{BP}

מתוצאות הטבלה נובע שקידוד בעזרת עץ סינטטי משפר את יכולת הדחיסה של המימושים. כמו כן נבדוק בתמונה מס' 1 $(\bar{P}_o = 0.71)$ את השתנות ערכי CR_{BP} כפונקציה של עצי פיסוק סינטטיים שונים; כשקידוד כל מישור סיבית נעשה על ידי עץ אחד ויחיד השייך למישור הסיבית המשמעותית ביותר. תוצאות הבדיקה מוצגות בטבלה הבאה:

טבלה מס' 3.10: ערכי CR_{BP} המתקבלים על ידי עצים שונים עבור תמונה מס' 1
$(\bar{P}_o = 0.71)$ (עץ אחד ויחיד לכל מישורי הסיבית).

Table 3.10: CR_{BP} values received with different parsing trees for picture No. 1 (One tree for all bit planes).

עץ A - סינטטי סטציונרי $P_o = 0.9$	עץ B - סינטטי סטציונרי $P_o = 0.70$	עץ D - סינטטי לא סטציונרי $\bar{P}_o = 0.8$	עץ II - תמונת פנים $\bar{P}_o = 0.63$	
0.94	1.0	0.97	1.0	CR_{BP}

תוצאות הטבלה מראות אף הפעם שעץ פיסוק סינטטי אוניברסלי יעיל גם כאשר משתמשים בעץ אחד עבור 4 מישורי הסיבית.

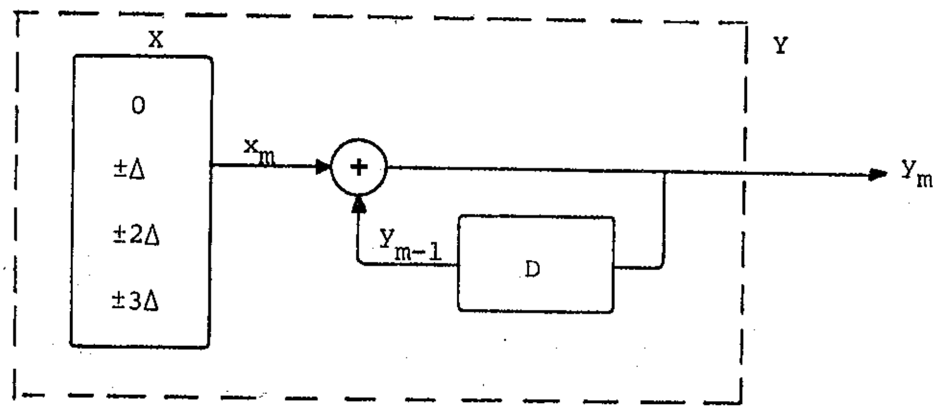
פרק 4 : הערכת ביצועי מימוש 4BP-ZL בעזרת מודל ליצירת תמונה

כאשר הועלה הרעיון להשתמש ביצוג תמונות לפי קוד GRAY כדי לתרגם את הקורלציה בין פיקסלים סמוכים לרצפים ארוכים יותר של אותו סימבול בינרי במישורי סיביות [13], נשאלה השאלה האם ניתן להעריך את יכולת הדחיסה של אלגוריתם זיו-למפל כאשר הוא מופעל על כל מישור סיבית לחוד.

על מנת להשיב לשאלה הזדקקנו למודל יצירת תמונה, אותו נציע בפרק זה [9]. בעזרת המודל נשחזר את התוכנות הסטטיסטיות של התמונה המשפיעות על ביצועי האלגוריתם. ננתח את המודל באופן אנליטי ונבדוק את התאמתו לתמונות אמיתיות מבחינת יכולת הדחיסה של האלגוריתם (על פי מימוש 4BP-ZL). כדי להעריך את טיב המודל נשווה את ביצועיו לאלה של מודלים אחרים.

4.1 מודל יצירת תמונה

תאור תהליך יצירת תמונה נעשה בעזרת המודל הבא:



ציור מס' 4.1: מודל ליצירת תמונה.

Fig. 4.1: Picture production model.

$$y_{m+1} = (y_m + x_{m+1})_{\text{mod } 256} \qquad m = 0,1,2,\ldots \tag{4.1}$$

כאשר y_m : ערך הפיקסל ה-m-י.

x_m : ערך הסימבול ה-m-י של המקור X.

4.1.1 איפיוני המקור X

המקור X פולט משתנים אקראיים X_1, X_2, X_3,... בלתי תלויים ובעלי התפלגות זהה. ניתן להתיחס למקור X כאל מקור אשר פולט סימבולים השייכים לתמונת הפרשים, זאת אומרת תמונה אשר נוצרת על ידי הפרש בין פיקסלים סמוכים ($x_m = y_m - y_{m-1}$). האלפאבית של המקור X נתון על ידי:

$$X = \{0,\Delta,-\Delta,2\Delta,-2\Delta,3\Delta,-3\Delta\}$$

גודל צעד הקואנטיזציה Δ נתון על ידי:

$$\Delta = 256/2^k \qquad (4.2)$$

כאשר k הוא מספר הסיביות ליצוג Y.

במקרה בו אנו דנים $k = 4$ לכן $\Delta = 16$.

הסתברות ההופעה של $x_m = a$, $a \in X$ שווה ל:

$$P_r\{x_m = 0\} = P_o \;;\; P_r\{x_m = \pm i\Delta\} = P_i \quad i = 1,2,3...$$

ההנחה שהמ"א x_m, $m = 1,2,...$ הם בלתי תלויים מסתמכת על העובדה שהקורלציה בין ערכי פיקסלים סמוכים בתמונת הפרש היא נמוכה ($\rho \approx 0.01 \div 0.3$).

4.1.2 קביעת ההסתברות להופעת הסימבולים של המקור X

קביעת ערכי ההסתברות P_0, P_1, P_2 ו-P_3 לסימבולים של המקור X נעשית על פי ערכי הממוצעים של $P_r\{x = a\}$ $a \in X$ שנמצאו ב-3 תמונות הפרשים של תמונות פנים.

יש לשים לב לנקודות הבאות:

1) לא נבדק בין אילו מצבים מתקיימים המעברים $\pm i\Delta$, $i = 0,1,2,3$, ולכן אנו מודדים את ההסתברות הממוצעת של המעבר בין המצבים השונים. תוצאות המדידה הנ"ל מופיעות בטבלה מס' 4.1.

2) אי הסטציונריות של התמונות גורמת לכך שערכי ההסתברות המתקבלים במדידה אינם מדויקים.

טבלה מס' 4.1: ערכי הסתברות הופעת הסימבולים של המקור X ב-3 תמונות פנים.

Table 4.1: Probability of the X source symbols for 3 face pictures.

תמונה	$P_r\{x=0\}$	$P_r\{x=\Delta\}$	$P_r\{x=-\Delta\}$	$P_r\{x\ 2\Delta\}$	$P_r\{x=-2\Delta\}$	$P_r\{x=\ 3\Delta\}$	$P_r\{x=-3\Delta\}$
1	0.714	0.117	0.128	0.014	0.012	0.0036	0.0035
2	0.66	0.148	0.162	0.008	0.012	0.0014	0.0027
3	0.719	0.121	0.140	0.004	0.008	0.0001	0.0014
ממוצע	0.698	0.129	0.143	0.009	0.011	0.0017	0.0025

על סמך המדידות שבטבלה נקבעים ערכי ההסתברות באופן הבא:

$$P_o = \langle P_r\{x = 0\}\rangle \qquad (4.3)$$

$$P_i = (\langle P_r\{x = i\Delta\}\rangle + \langle P_r\{x = -i\Delta\}\rangle)/2 \qquad i = 1,2,3$$

כדי שסכום הסתברויות ההופעה של כל הסימבולים במקור X יהיה שווה ל-1 נקבע את P_3 על ידי:

$$P_3 = (1-(P_0+2P_1+2P_2))/2$$

הערכים המספריים עבור מודל זה של המקור X הם:

$P_3 = 0.0053$, $P_2 = 0.0097$, $P_1 = 0.136$, $P_0 = 0.698$.

אנטרופית המקור X היא:

$$H(x) = -\sum_{i=1}^{3} 2P_i \log P_i - P_0 \log P_0 \qquad (4.4)$$

4.1.3 איפיוני המקור Y

טענה מס' 1: Y הינו מקור מרקובי מסדר ראשון בעל האלפאבית הבא:

$$Y = \{0,\Delta,2\Delta,\ldots,(2^k-1)\Delta\}$$

הוכחה

$$P_r\{y_{m+1}=j/y_0,y_1,\ldots,y_m\} = P_r\{x_{m+1} = (j-y_m) \bmod 256/y_0,y_1,\ldots y_m\}$$

$$= P_r\{x_{m+1} = (j-y_m) \bmod 256\} = P((j-y_m) \bmod 256)$$

מפני ש-x_{m+1} בלתי תלוי ב-y_0 , $y_1,\ldots,$, y_m ;

לכן:

$$P(i,j) \triangleq P_r\{y_{m+1} = j/y_m = i\} = P((j-i) \bmod 256)$$

כאשר $j, i \in Y$.

מ.ש.ל.

<u>מסקנה</u>: סדרת המשתנים האקראיים y_m מהווה שרשרת מרקובית. הרכב הא"ב של Y נגזר מתוך תהליך בנית התמונה, על ידי סיכום הסימבולים של המקור X במדולו 256. בסיכום זה מובטח שערכי הפיקסלים נשארים בתחום הגדרתם. עבור $k = 4$ הא"ב של Y מורכב מהסימבולים הבאים:

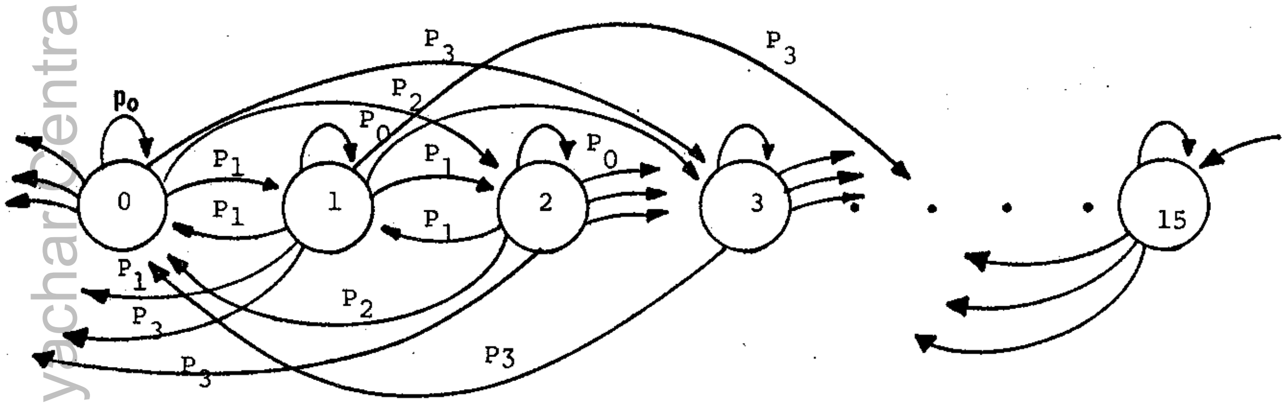
$$Y = \{0,16,32,\ldots.,240\}$$

מטריצת המעבר P של שרשרת מרקובית y_m עבור $k = 4$ היא.

$$P = \begin{bmatrix}
P_0 & P_1 & P_2 & P_3 & & & & & & & & & & P_3 & P_2 & P_1 \\
P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & & & & & P_3 & P_2 \\
P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & & & & & P_1 \\
P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & & & & \\
& P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & & & \\
& & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & & \\
& & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & & \\
& & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & & \\
& & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & & \\
& & & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & & \\
& & & & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & & \\
& & & & & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 & \\
& & & & & & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 & P_3 \\
P_3 & & & & & & & & & & P_3 & P_2 & P_1 & P_0 & P_1 & P_2 \\
P_2 & P_3 & & & & & & & & & & P_3 & P_2 & P_1 & P_0 & P_1 \\
P_1 & P_2 & P_3 & & & & & & & & & & P_3 & P_2 & P_1 & P_0
\end{bmatrix} \qquad (4.5)$$

כאשר הסתברות המעבר ממצב i למצב j היא $P(i,j)$, מתקבלת מטריצה מרקובית כפולה; הווה אומר סכום כל עמודה וכל שורה שווה ל-1.

דיאגרמת המצבים עבור שרשרת מרקובית y_m היא:



ציור מס' 4.2: דיאגרמת המצבים עבור שרשרת מרקוב המתארת את המקור Y.
Fig. 4.2: State Diagram of The Y source Markov Chain.

כדי לפשט את ניסוח הבעיה נסמן את מצבי השרשרת y_m על ידי:

$$e = \{0,1,...,15\}$$

במקום על ידי סימבולי הא"ב של Y.

טענה מס' 2: במצב המתמיד של השרשרת המרקובית Y הסתברות ההופעה של כל מצב היא:

$$j \in \{0,1,...,15\}, \qquad \pi(j) = \frac{1}{16}$$

הוכחת הטענה בנספח ז'.

4.1.4 אנטרופיה של המקור Y

המקור Y, על פי טענת מס' 1, הוא מקור מרקובי מסדר ראשון, לכן אנטרופית המקור Y היא:

$$H(y_m/y_{m-1}) = -\sum_{i=0}^{2^k-1} P_i(y_m,y_{m-1}) \log P_i(y_m/y_{m-1}) \tag{4.6}$$

<u>טענה מס' 3:</u> אנטרופיה מסדר ראשון של המקור Y שווה לאנטרופיה מסדר אפס של המקור X עבור המודל הנ"ל, $H(y_m/y_{m-1}) = H(x)$.
הוכחת הטענה מופיעה בנספח ח' .

4.2 <u>אנטרופיה של מישור סיבית של המקור Y</u>

לצורך חישוב האנטרופיה של מישור סיבית ℓ , $\ell=0,1,2,3$, של המקור Y נגדיר את המשתנים האקראיים $B_\ell(i)$, כאשר i הוא מצב בשרשרת המרקובלית y_m ו-ℓ הוא מישור סיבית של היצוג הבינרי או היצוג לפי קוד GRAY של אותם מצבים. המ"א $B_\ell(i)$ מקבלים את הערכים 0 או 1, ולכן נתיחס אליהם כאל מקורות בינריים. עבור מצב i נתון נסמן ב-j_k, $j_k \in e$, לכל j_k, את המצבים בהם הסתברות המעבר מ-i ל-j_k שונה מאפס, וכמו כן עבור ℓ ו-i נתון נסמן ב-$P^i_\ell(0)$ וב-$P^i_\ell(1)$ את ההסתברות שה"מקור" $B_\ell(i)$ "יפלוט" את הסימבול '0' או '1' בהתאמה. ההסתברות שה"מקור" $B_\ell(i)$ "יפלוט" '0' או '1' תלויה ב:

1) הסתברות המעבר של המצב ה-i-י למצבים j_1, j_2, ..., j_k, ..., $j_k \in e$.
2) היצוג הבינרי או לפי קוד GRAY של ערכי המצבים j_1, j_2, ..., j_k, ...

האנטרופיה של ה"מקור" $B_\ell(i)$, עבור המצב ה-i-י והסיבית ה-ℓ-ית נתונה על ידי הביטוי:

$$H_i(B_\ell) = -P^\ell_i(0)\log P^\ell_i(0) - P^\ell_i(1)\log P^\ell_i(1) \qquad i = 0,1,\ldots,15 \tag{4.8}$$

האנטרופיה הממוצעת למישור סיבית ℓ כלשהיא נתונה על ידי הביטוי:

$$H(Y_\ell) = \sum_{i=0}^{15} P(Y=i)\, H_i(B_\ell) \tag{4.9}$$

לפי טענה מס' 3 ההסתברות להופעת מצב i היא $P(Y=i) = 1/16$ עבור כל i ,
לכן נרשום את הביטוי (4.9) בצורה הבאה:

$$H(Y_\ell) = 1/16 \sum_{i=0}^{15} H_i(B_\ell) \tag{4.10}$$

הדחיסה הכוללת הניתנת להשגה על ידי שיטת קידוד זו (כל מישור סיבית לחוד) נתונה על ידי:

$$H(Y) = \sum_{\ell=0}^{3} H(Y_{\ell}) \qquad (4.11)$$

4.2.1 אנטרופיה עבור ייצוג GRAY של המקור Y

נחשב כעת את האנטרופיה עבור 4 מישורי הסיבית של ייצוג GRAY של ערכי המצבים.

שלבי החישוב:

1) רישום של ערכי המ"א $B_{\ell}(i)$ וההסתברויות לקבלם.

דוגמא: עבור הסיבית המשמעותית ביותר, $\ell = 0$, והמצב ההתחלתי $i = 0$ נקבל:

טבלה 4.2: הסתברויות המעבר וערכי $B_0(0)$ עבור מישור סיבית $\ell = 0$ ומצב התחלתי $i = 0$.

Talbe 4.2: Transition probabilities and $B_0(0)$ values for bit plane $\ell = 0$ with initial state $i = 0$.

מצב j_k	הסתברות המעבר (מהמצב $i = 0$)	ערך $B_0(0)$
0	P_0	0
1	P_1	0
2	P_2	0
3	P_3	0
15	P_1	1
14	P_2	1
13	P_3	1

2) חישוב ההסתברות הכוללת לפליטת 0, $P_i^{\ell}(0)$ ו-1 $P_i^{\ell}(1)$ של $B_{\ell}(i)$.

עבור $\ell = 0$ ו-$i = 0$ ה"מקור" הוא $B_0(0)$ וההסתברויות לפליטת 0 ו-1 הן:

$$P_0^0(1) = P_1 + P_2 + P_3$$

$$P_0^0(0) = P_0 + P_1 + P_2 + P_3 = 1 - (P_1 + P_2 + P_3)$$

בטבלה מס' 4.3 מופיעות ההסתברויות לפליטת 1, $P_i^\ell(1)$, של $B_\ell(i)$ לכל מצב i עבור ייצוג GRAY של ערכי המצבים.

ההסתברות לפליטת 0, $P_i^\ell(0)$ נתונה על ידי $P_i^\ell(0) = 1 - P_i^\ell(1)$.

3) חישוב האנטרופיה עבור כל מישור סיבית.

א. אנטרופיה של המקורות $B_\ell(i)$ לכל i ו-ℓ.

חישוב זה מתבסס על נוסחאות מס' (4.8). לדוגמא, עבור $\ell = 0$, $i = 0$ נקבל:

$$H_0(B_0) = -(P_1+P_2+P_3)\log(P_1+P_2+P_3) - (1-(P_1+P_2+P_3))\log(1-(P_1+P_2+P_3)) \tag{4.12}$$

כאשר $H_0(B_0)$ מסמן את האנטרופיה של המקור $B_0(i)$ עבור המצב $i = 0$.

חישוב האנטרופיה עבור $\ell = 0$ לכל i.

במצבים $i = 0,7,8,15$ מתקבל ביטוי (4.12).

במצבים $i = 1,6,9,14$ אנטרופית המקור $B_0(i)$ נתונה על ידי:

$$H_i(B_0) = -(P_2+P_3)\log(P_2+P_3)-(1-(P_2+P_3))\log(1-(P_2+P_3)) \tag{4.13}$$

במצבים $i = 2,5,10,13$ אנטרופית המקור $B_0(i,j)$ נתונה על ידי:

$$H_i(B_0) = -P_3 \log P_3 - (1-P_3) \log (1-P_3) \tag{4.14}$$

במצבי i, $i = 3,4,11,12$ האנטרופיה שווה לאפס, דהיינו:

$$H_i(B_0) = 0 \tag{4.15}$$

ב. האנטרופיה הממוצעת עבור כל מישור סיבית.

חישוב זה מתבסס על נוסחה מס' (4.9).

האנטרופיה הממוצעת שמתקבלת עבור מישור **הסיבית** $\ell = 0$ היא:

$$H(Y_0) = \sum_{i=0}^{15} P_r(Y=i)\, H_i(B_0) = \frac{4}{16}\,(H_0(B_0) + H_1(B_0) + H_2(B_0)) \tag{4.16}$$

בדרך דומה ועל סמך התוצאות שמופיעות בטבלה מס' 4.3 נחשב את האנטרופיה הממוצעת שמתקבלת עבור מישור **הסיבית** $\ell = 0$ שהיא:

טבלה מס' 4.3: הסתברויות לפליטת סימבולים של "המקורות" $B_\ell(i)$.

Table 4.3: Probability of the $B_\ell(i)$ "sources" symbols.

מצב	$P_i^0(1)$ ($\ell=0$)	$P_i^1(1)$ ($\ell=1$)	$P_i^2(1)$ ($\ell=2$)	$P_i^3(1)$ ($\ell=3$)
0	$P_1+P_2+P_3$	0	P_2+2P_3	$P_1+2P_2+P_3$
1	P_2+P_3	P_3	$P_1+P_2+P_3$	$1-(P_1+2P_2+P_3)$
2	P_3	P_2+P_3	$1-(P_1+P_2+P_3)$	$1-(P_1+2P_2+P_3)$
3	0	$P_1+P_2+P_3$	$1-(P_2+2P_3)$	$P_1+2P_2+P_3$
4	0	$1-(P_1+P_2+P_3)$	$1-(P_2+2P_3)$	$P_1+2P_2+P_3$
5	P_3	$1-(P_2+P_3)$	$1-(P_1+P_2+P_3)$	$1-(P_1+2P_2+P_3)$
6	P_2+P_3	$1-P_3$	$P_1+P_2+P_3$	$1-(P_1+2P_2+P_3)$
7	$P_1+P_2+P_3$	1	P_2+2P_3	$P_1+2P_2+P_3$
8	$1-(P_1+P_2+P_3)$	1	P_2+2P_3	$P_1+2P_2+P_3$
9	$1-(P_2+P_3)$	$1-P_3$	$P_1+P_2+P_3$	$1-(P_1+2P_2+P_3)$
10	$1-P_3$	$1-(P_2+P_3)$	$1-(P_1+P_2+P_3)$	$1-(P_1+2P_2+P_3)$
11	1	$1-(P_1+P_2+P_3)$	$1-(P_2+2P_3)$	$P_1+2P_2+P_3$
12	1	$P_1+P_2+P_3$	$1-(P_2+2P_3)$	$P_1+2P_2+P_3$
13	$1-P_3$	P_2+P_3	$1-(P_1+P_2+P_3)$	$1-(P_1+2P_2+P_3)$
14	$1-(P_2+P_3)$	P_3	$P_1+P_2+P_3$	$1-(P_1+2P_2+P_3)$
15	$1-(P_1+P_2+P_3)$	0	P_2+2P_3	$P_1+2P_2+P_3$

$$H(Y_1) = \frac{4}{16}\,(H_1(B_1) + H_2(B_1) + H_3(B_1)) \qquad (4.17)$$

כאשר $H_1(B_1)$ היא האנטרופיה של המצבים $i = 1,6,9,14$ והיא שווה ל:

$$H_1(B_1) = -P_3 \log P_3 - (1-P_3) \log(1-P_3)$$

$H_2(B_1)$ היא האנטרופיה של המצבים $i = 2,5,10,13$ והיא שווה ל:

$$H_2(B_1) = -(P_2+P_3) \log(P_2+P_3) - (1-(P_2+P_3)) \log(1-(P_2+P_3))$$

$H_3(B_1)$ היא האנטרופיה של המצבים $i = 3,4,11,12$ והיא שווה ל:

$$H_3(B_1) = -(P_1+P_2+P_3) \log(P_1+P_2+P_3) - (1-(P_1+P_2+P_3)) \log(1-(P_1+P_2+P_3))$$

במצבים $i = 0,7,8,15$ האנטרופיה שווה לאפס.

במישור סיבית $\ell = 2$ האנטרופיה הממוצעת היא:

$$H(Y_2) = \frac{8}{16}\,(H_0(B_2) + H_1(B_2)) \qquad (4.18)$$

כאשר $H_0(B_2)$ היא האנטרופיה של המצבים $i = 0,3,4,7,8,11,12,15$ והיא שווה ל- $H_0(B_2) = -(P_2+2P_3) \log(P_2+2P_3) - (1-(P_2+2P_3)) \log(1-(P_2+2P_3))$.

$H_1(B_2)$ היא האנטרופיה המתקבלת במצבים $i = 1,2,5,6,9,10,13,14$ והיא נתונה על ידי:

$$H_1(B_2) = -(P_1+P_2+P_3) \log(P_1+P_2+P_3) - (1-(P_1+P_2+P_3)) \log(1-(P_1+P_2+P_3))$$

במישור סיבית $\ell = 3$ האנטרופיה הממוצעת היא:

$$H(Y_3) = H_0(B_3) \qquad (4.19)$$

כאשר $H_0(B_3)$ היא האנטרופיה המתקבלת בכל המצבים $i = 0,1,\ldots,15$ והיא שווה ל:

$$H_0(B_3) = -(P_1+2P_2+P_3) \log(P_1+2P_2+P_3) - (1-(P_1+2P_2+P_3)) \log(1-(P_1+2P_2+P_3))$$

4.2.2 <u>האנטרופיה עבור היצוג הבינרי של המקור Y</u>

נחשב את האנטרופיה של המקור Y ביצוג הבינרי של ערכי המצבים בדרך בה חישבנו את האנטרופיה של Y ביצוג GRAY. בנספח (ט') מופיעה טבלת ההסתברויות לפליטת '0', $P_i^{\ell}(0)$ ו-'1', $P_i^{\ell}(1)$ של המקור Y עבור כל מישור סיבית ℓ, $\ell=0,1,2,3$, כאשר היצוג של ערכי המצבים הינו בינרי.

להלן תוצאות חישוב האנטרופיה הממוצעת, של כל מישור סיבית.

במישור סיבית $\ell = 0$ האנטרופיה הממוצעת היא:

$$H(Y_0) = \frac{4}{16}\,(H_0(B_0) + H_1(B_0) + H_2(B_0)) \tag{4.20}$$

כאשר:

$$H_0(B_0) = -P_3 \log P_3 - (1-P_3)\log(1-P_3)$$

$$H_1(B_0) = -(P_2+P_3)\log(P_2+P_3) - (1-(P_2+P_3))\log(1-(P_2+P_3))$$

$$H_2(B_0) = -(P_1+P_2+P_3)\log(P_1+P_2+P_3)-(1-(P_1+P_2+P_3))\log(1-(P_1+P_2+P_3))$$

במישור סיבית $\ell = 1$ האנטרופיה הממוצעת היא:

$$H(Y_1) = \frac{8}{16}\,(H_0(B_1) + H_1(B_1)) \tag{4.21}$$

כאשר:

$$H_0(B_1) = H_2(B_0)$$

$$H_1(B_1) = -(P_2+2P_3)\log(P_2+2P_3)-(1-(P_2+2P_3))\log(1-(P_2+2P_3))$$

במישור סיבית $\ell = 2$ האנטרופיה הממוצעת היא:

$$H(Y_2) = H_0(B_2) = -(P_1+2P_2+P_3)\log(P_1+2P_2+P_3)- \\ -(1-(P_1+2P_2+P_3))\log(1-(P_1+2P_2+P_3)) \tag{4.22}$$

במישור סיבית $\ell = 3$ האנטרופיה הממוצעת היא:

$$H(Y_3) = H_0(B_3) = -(2P_1+2P_3)\log(2P_1+2P_3)- \\ -(1-(2P_1+2P_3))\log(1-(2P_1+2P_3)) \tag{4.23}$$

התבוננות בנוסחאות האנטרופיה של מישורי הסיבית בשני היצוגים מצביעה על קיום התכונות הבאות:

א) נוסחת האנטרופיה של מישור הסיבית $\ell = 0$ (MSB) שווה בשני היצוגים, כלומר $H_{GRAY}(Y_0) = H_{BINARY}(Y_0)$.

ב) נוסחת האנטרופיה של מישור הסיבית $\ell = 1$ ביצוג GRAY שווה לזו של $\ell = 0$ באותו יצוג, כלומר $H_{GRAY}(Y_0) = H_{GRAY}(Y_1)$.

ג) נוסחאות האנטרופיה של מישורי הסיבית $\ell = 2$ ו-$\ell = 3$ ביצוג ה-GRAY שוות לאלה של היצוג הבינרי עבור מישורי הסיבית $\ell = 1$ ו-$\ell = 2$ בהתאמה, כלומר $H_{GRAY}(Y_2) = H_{BINARY}(Y_1)$ ו-$H_{GRAY}(Y_3) = H_{BINARY}(Y_2)$.

ד) נוסחת האנטרופיה של מישור הסיבית $\ell = 3$ (LSB) ביצוג הבינרי יחודית משום שרק במישור סיבית זה מתקבל רצף של אפס ואחד לסרוגין.

ה) אם נייצג את מקור Y על ידי יותר מ-4 סיביות נקבל ש:

1) האנטרופיה של 3 מישורי הסיבית הפחות משמעותיות ביותר נשארת ללא שינוי בשני היצוגים.

2) ביצוג GRAY האנטרופיה של מישורי הסיבית פרט ל-3 הפחות משמעותיות ביותר שווה לערך האנטרופיה של מישורי הסיבית $\ell = 0$ כשהמקור מוצג על ידי 4 סיביות; זאת משום שהרצפים של אפסים ואחדים במישורי סיבית אלה נעשים ארוכים יותר ולכן פוחת ערך האנטרופיה של כל מישור סיבית לחוד. מכאן שהאנטרופיה הממוצעת הכוללת של המקור Y ביצוג GRAY נשארת קבועה.

3) ביצוג הבינרי ערך האנטרופיה הממוצעת הכוללת הולך וגדל ככל שמשתמשים במספר סיביות גדול יותר ליצוג המקור Y.

4.3 ניתוח המודל

עד כה הוצג מודל יצירת התמונה. כעת נבדוק את תכונותיו ואת התאמתו לתמונות אמיתיות. בדיקת תכונותיו והתאמתו מתבססת על ערכי האנטרופיה שמתקבלים באמצעות המודל עבור הפרמטרים השונים.

4.3.1 תכונות המודל

לאחר הצבת הערכים של הסתברות הופעת הסימבולים של המקור X, (P_3,P_2,P_1,P_0), בביטויים שהתקבלו עבור האנטרופיה של כל מישור סיבית של המקור Y, ביצוגו הבינרי וביצוגו לפי קוד GRAY מתגלה כי:

1) בשני היצוגים של המקור מתקיים:

$$H_0(B) \leq H_1(B) \leq H_2(B) \leq H_3(B)$$

ההסבר לכך הוא כדלקמן: מישורי הסיביות היותר משמעותיות מיצגים יותר את המבנה הסטטיסטי של התמונה מאשר מישורי הסיביות הפחות משמעותיות. ולכן, הראשונים הן בעלי מבנה מוגדר יותר. בתמונות קימת קורלציה גבוהה בין פיקסלים סמוכים, תכונה זו מתורגמת במישורי הסיבית המשמעותיות יותר לרצף ארוך יותר של אותו סימבול, לכן אי הודאות לקבל "0" או "1" נמוכה. ככל שהמשקל של מישור הסיבית בערך הפיקסל הוא נמוך יותר חוסר הקורלציה של הסיביות באותו מישור גדלה ומידת הודאות לקבלת 0 או 1 פוחתת. זאת משום שתרומתו של מישור הסיבית לערך הפיקסל פוחתת. כיוון שהאנטרופיה היא מדד לאי הודאות מתקיימת התכונה לעיל.

2) ביצוג GRAY מתקבלת דחיסה גבוהה יותר מאשר ביצוג הבינרי.
עבור כל מישור סיבית מתקיים:

$$\ell = 0,1,2,3, \qquad H_{GRAY}(Y_\ell) \leq H_{BINARY}(Y_\ell)$$

ולכן אם מסכמים את האנטרופיה של כל המישורים מתקיים:

$$H_{GRAY}(Y) < H_{BINARY}(Y)$$

השוויון לא מתקיים כי במישורים $\ell = 1,2,3$ מתקבל אי שוויון מוחלט:

$$H_{GRAY}(Y_\ell) < H_{BINARY}(Y_\ell)$$

מכיוון שיצוג GRAY של ערכי המצבים מנצל טוב יותר את הקורלציה הקיימת בין ערכי הפיקסלים סמוכים. כדי להמחיש טענה זו נציג את הדוגמא הבאה:
תהיה נתונה שרשרת מרקוב כפי שתואר במודל, שבה ההסתברות למעברים בין המצבים 7 ו-8 וחזרה שווה בקירוב להסתברות להשאר באותו מצב ($P_0 \simeq 0.5$).
ביצוג הבינרי של ערכי המצבים נקבל: B# 7 = 0111, B# 8 = 1000; וביצוג GRAY: G# 7 = 0100, G# 8 = 1100.
במקרה של דוגמא זו ועל סמך המודל נקבל שביצוג GRAY ערכי המ"א $B_\ell(7)$ ו-$B_\ell(8)$ לכל ℓ, פרט ל-$\ell = 0$, נשארים קבועים, ואילו ביצוג הבינרי ערכי המ"א הנ"ל משתנים בכל מעבר ממצב למצב. מכאן שההסתברות ש"המקורות" $B_\ell(7)$ ו-$B_\ell(8)$ עבור $\ell = 0,1,2,3$ יפלטו את אותו סימבול ("0" או "1") גבוהה יותר (ולכן האנטרופיה נמוכה יותר) ביצוג לפי GRAY מאשר ביצוג הבינרי.

3) האם מספיקות 4 סיביות כדי לייצג את המקור Y ?
התשובה לשאלה זו מתחלקת לשניים.

א. מבחינת איכות התמונה הובהר בסעיף 1.5 שהאיכות המתקבלת על ידי 4 סיביות לפיקסל עבור תמונות פנים טובה בהחלט.

ב. מבחינת המודל עצמו, הראנו שכאשר למקור X יש 7 סימבולים, ערך האנטרופיה הממוצע הכולל ביצוג GRAY $H_{GRAY}(Y)$ (נוסחה (4.11)) נשאר קבוע כאשר מספר הסיביות בהן משתמשים ליצוג המקור Y גדול מ-4. מכאן שכאשר למקור X שבמודל ישנם 7 סימבולים, השימוש במודל מוגבל לתאור תמונות המיוצגות על ידי 4 סיביות לפיקסל.

4) האם תמיד עדיף יצוג לפי קוד GRAY על פני היצוג הבינרי למודל זה?
התוצאות שתוארו עד כאן מתקיימות עבור תחום רחב של הסתברויות הסימבולים $\pm i\Delta$, $i = 0,1,2,3$ אך יתכן שבמקורות בהם ההסתברות של אותם סימבולים רחוקה

מהערכים שקיבלנו עבור תמונות פנים לא מתקיים אי-השוויון $H_{GRAY}(Y)<H_{BINARY}(Y)$.

לדוגמא עבור המקרה בו $P_0 \simeq 0$, $P_1 = 0.35$, $P_2 = 0.05$, $P_3 = 0.10$

מתקבל ש-$H_{GRAY}(Y)>H_{BINARY}(Y)$ ו-$H_{GRAY}(Y) - H_{BINARY}(Y) = 0.0507$.

במודלים פשוטים יותר ליצירת תמונה, אותם נתאר בסוף הפרק ניתן יהיה לראות באופן ברור יותר באלה מקרים מתקיים אי השוויון $H_{GRAY}(Y) > H_{BINARY}(Y)$.

4.4 התאמת המודל לתמונות אמיתיות

נבדוק את התאמת המודל לתמונות אמיתיות על ידי השוואת ערכי האנטרופיה של התמונות ושל המודל. בנוסחאות שהתקבלו בפיתוח המודל נציב את ערכי ההסתברות של הסימבולים של המקור X המופיעים בסעיף 4.1.2. בנוסף לערכי אנטרופית המודל נציג את ערכי האנטרופיה והדחיסה המתקבלים מתמונה סינטטית שנבנתה על פי המודל. תוצאות אלה מופיעות בטבלאות השונות תחת הכותרת "תמונה סינטטית".

4.4.1 השוואת האנטרופיה המותנית

מודל יצירת התמונה מיוצג על ידי המקור Y, שהוא מקור מרקובי מסדר ראשון, אשר באמצעותו אנו מנסים לתאר את התמונה האמיתית. המקור Y מכיל את המקור X של סימבולים בלתי תלויים אשר מייצג את תמונת הפרשים של התמונה האמיתית.

עבור מקורות מרקוביים מסדר ראשון, הוכח בטענה מס' 3 (עמ' 73) שהאנטרופיה המותנית של המקור Y שווה לאנטרופיה מסדר אפס, שלהמקור X, $H(y_n/y_{n-1}) = H(X)$.

בטבלה הבאה נציג את ערכי האנטרופיה המותנית $H(y_n/y_{n-1})$ ואת ערכי האנטרופיה מסדר אפס $H(x)$ של תמונות פנים, של המודל ושל התמונה הסינטטית. האנטרופיה $H(x)$ נמדדת על תמונות הפרשים.

טבלה מס' 4.4: ערכי האנטרופיה המותנית של תמונות פנים של המודל וערכי האנטופיה מסדר אפס של תמונות הפרשים ושל המודל.

Table 4.4: Conditional Entropy values for face pictures and model; Zero Order Entropy for difference pictures and model.

תמונה	$H(y_n/y_{n-1})$	$H(x)$
1	1.2986	1.3863
2	1.3382	1.4451
3	1.0560	1.2594
ממוצע	1.2309	1.3636
מודל	1.3548	1.3548
ת. סינטטית	1.3554	(1)1.3554

(1) הערה: ההבדל הקטן בין ערכי האנטרופיה של המודל שחושבה על פי נוסחאות לבין ערכי האנטרופיה שנמדדה בתמונה סינטטית נובע מהגודל הסופי של התמונה.

מתוך התוצאות המוצגות בטבלה הנ"ל ניתן לראות שמבחינת האנטרופיה, המודל המבוסס על מקור מרקובי מסדר ראשון מתאר בצורה סבירה את התכונות הסטטיסטיות של תמונות הפנים. ההפרשים בערכי האנטרופיה בתמונות 1 ו-2 הם קטנים ואילו בתמונה מס' 3 ההפרש משמעותי. לכן, לתאור תמונה 3 נדרש מודל מורכב יותר בדומה למודל המבוסס על שירשור של 2 מקורות מרקוביים מסדר ראשון (ראה סעיף 4.5.3).

4.4.2 השוואת תוצאות הדחיסה של מישורי הסיבית

השוואת תוצאות הדחיסה של התמונות עם ערכי האנטרופיה של המודל מתבססת על משפט מס' 3 , אשר מופיע במאמר[1] של זיו ולמפל. במשפט נטען ש-$\rho(x)$, מספר הסיביות לאות מקור המתקבל לאחר הדחיסה לפי האלגוריתם ZL, שואף לאנטרופית המקור ($\rho(x) \to H(x)$); ראה סעיף 2.1.1.
בטבלאות מס' 4.5 ו-4.6 מופיע מספר הסיביות לפיקסל $\rho(u)$ המתקבל עבור 4 הסיביות המשמעותיות ביותר של קבוצת תמונות פנים בייצוג הבינרי ובייצוג GRAY של התמונות ושל המודל $H(Y)$. תוצאות הדחיסה עבור כל מישור סיבית

מתקבלות באמצעות אלגוריתם זיו-למפל במימוש של עץ פיסוק בינרי רגיל (סעיף 2.2.3). בטבלאות מופיע גם הערך $\hat{H}(u)$, משערך האנטרופיה לפי אלגוריתם ZL, שהוא החסם התחתון למספר הסיביות לפיקסל שניתן לקבל באמצעות האלגוריתם.

טבלה מס' 4.5: ערכי הדחיסה של כל מישור סיבית ביצוג הבינרי של המקור, $\rho_{4BP}(u)$: מספר הסיביות לפיקסל ו-$\hat{H}(u)$: משערך האנטרופיה.

Table 4.5: Comparisson values for each bit plane for binary representation of the source, also $\rho_{4BP}(u)$ and $\hat{H}(u)$ values.

תמונה	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\rho_{4BP}(u)$ (*)	$\hat{H}(u)$ (*)
1	0.3037	0.4815	0.6720	0.8561	2.3133	1.3854
2	0.2140	0.3894	0.6817	0.9180	2.2031	1.5192
3	0.2271	0.3435	0.5482	0.8010	1.9198	1.2525
ממוצע	0.2483	0.4048	0.6340	0.8584	2.1455	1.3857
מודל	0.1931	0.3777	0.6360	0.8590	2.0658 (2)	1.3548 (1)
ת. סינטטית	0.3180	0.5169	0.7385	0.9460	2.5194	1.6895

(*) $\frac{\text{סיביות}}{\text{פיקסל}}$

הערות

(1) הערך של $\hat{H}(u)$ עבור המודל הינו הערך המחושב של אנטרופית המקור Y ($H(y_n/y_{n-1})$).

(2) הערך המחושב של $\rho_{4BP}(u)$ של המודל הינו $H(Y)$.

<u>טבלה מס' 4.6</u>: ערכי הדחיסה של כל מישור סיבית ביצוג ה-GRAY של המקור.

<u>Table 4.6</u>: Compression values for each bit plane for GRAY code representation of the source.

תמונה	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\rho_{4BP}(u)$ (*)	$\hat{H}(u)$ (*)
1	0.3038	0.3031	0.4876	0.6392	1.7337	1.3854
2	0.2140	0.3056	0.5117	0.6498	1.6811	1.5192
3	0.2271	0.2212	0.3852	0.6013	1.4348	1.2525
ממוצע	0.2483	0.2767	0.4615	0.6301	1.6166	1.3857
מודל	0.1981	0.1931	0.3777	0.6360	1.3999 (2)	1.3548 (1)
ת. סינטטית	0.3180	0.3160	0.5187	0.7389	1.8916	1.6895

(*) סיביות לפיקסל.

<u>הערות</u>

(1) ו-(2) , ראה הערות (1) ו-(2) המוזכרות לעיל.

מתוך התוצאות שהוצגו בטבלאות הנ"ל ניתן להסיק את המסקנות הבאות:

1. תוצאות הדחיסה המתקבלות על ידי קידוד התמונות ביצוג ה-GRAY טובות יותר מאשר אלה שמתקבלות על ידי קידוד ביצוג הבינרי של אותן תמונות. התוצאות מתאימות לאלה שהתקבלו עבור מודל יצירת התמונה.

2. החפרש בין ערכי האנטרופיה הכוללת של 4 מישורי הסיבית של מודל התמונה לבין מספר הסיביות לפיקסל המתקבל לאחר דחיסת תמונות פנים לפי מימוש 4BP-ZL של האלגוריתם זיו-למפל הינו 0.08 סיביות לפיקסל מתוך 2.06 סיביות לפיקסל ביצוג הבינרי של המקור. ביצוג GRAY של המקור ההבדל הוא 0.22 סיביות לפיקסל (מתוך 1.399 סיביות לפיקסל). למרות הבדלים אלה המודל מתאר באופן סביר, מבחינת האנטרופיה, את התמונה.

3. במישורי הסיבית היותר משמעותיות אין התאמה טובה במיוחד בייצוג GRAY של התמונות בין ערכי האנטרופיה של המודל לבין תוצאות דחיסת התמונות.

ניתן להסביר אה אי ההתאמה בתוצאות המוצגות בנקודות 2 ו-3 על ידי:

א) אי-הסטציונריות של התמונות, תכונה בה המודל לא מתחשב.

ב) התכנסות איטית של אלגוריתם זיו-למפל. התכנסות אלגוריתם ZL איטית יותר ככל שהמקור בעל אנטרופיה נמוכה יותר; מכאן, שאי התאמה בין ערכי האנטרופיה המחושבים על פי המודל לבין הערכים הנמדדים עבור התמונה הסינטטית רבה יותר במישורי הסיביות היותר משמעותיות.

ג) הנחה של מקור ללא זכרון - לגבי תמונות הפרשים - אינה מדויקת, מכיוון שקיימת קורלציה, אומנם קטנה, בין פיקסלים סמוכים בתמונות הפרשים. כמו כן לא נבדקה ההנחה שהסתברויות המעבר בין המצבים שוות. יתכן שקיימות סטיות קלות בערכי ההסתברות עבור המצבים השונים. מכאן שערכי ההסתברויות P_0, P_1, P_2 ו-P_3 שנמדדו בתמונות ההפרשים אינם מדויקים.

4.5 מודלים אחרים ליצירת תמונה

נשאלת השאלה האם המודל שהוצג הינו המודל היחיד ואם איננו יחיד האם הוא הטוב ביותר. מטרת המודל היא בעיקר להסביר מדוע יצוג GRAY טוב יותר מהיצוג הבינרי הרגיל ולהעריך את מידת הדחיסה שניתן להשיג על ידי הפעלת המימוש 4BP-ZL. בסעיף זה נבדוק מודלים נוספים אחרים פשוטים יותר מהמודל שתואר ואף נציג מודלים מסובכים יותר.

4.5.1 מודלים פשוטים:

נשווה את המודל שהוצג עם מודל זהה ליצירת תמונה שבו הא"ב של המקור X מצומצם יותר.

א. מודל בעל א"ב של 5 אותיות: $X = \{0,\Delta,-\Delta,2\Delta,-2\Delta\}$

על סמך ערכי ההסתברות המופיעים בטבלה מס' 4.1 נקבעים הערכים הבאים:

$$P_0 = P_r\ \{x=0\} \quad = 0.70$$

$$P_1 = P_r\ \{x=\pm\Delta\} \quad = 0.14$$

$$P_2 = P_r\ \{x=\pm 2\Delta\} \quad = 0.01$$

לאחר פיתוח המודל נקבל עבור א"ב של 5 אותיות את נוסחאות האנטרופיה הבאות;
עבור כל מישור סיבית ועבור יצוג GRAY של המקור:

$$H(Y_0) = \frac{4}{16}\,(H(P_1+P_2) + H(P_2)) \tag{4.24}$$

$$H(Y_1) = \frac{4}{16}\,(H(P_1+P_2) + H(P_2)) \tag{4.25}$$

$$H(Y_2) = \frac{8}{16}\,(H(P_1+P_2) + H(P_2)) \tag{4.26}$$

$$H(Y_3) = H(P_1+2P_2) \tag{4.27}$$

כאשר

$$H(P) \triangleq -P \log P - (1-P) \log (1-P) \tag{4.28}$$

האנטרופיה הכוללת נתונה על ידי:

$$H(Y) = \sum_{\ell=0}^{3} H(Y_\ell)$$

ב. מודל בעל א"ב של 3 אותיות:

$$X = \{0,\Delta,-\Delta\}$$

על סמך ערכי ההסתברות המופיעים בטבלה מס' 4.1 נקבע ש:

$$P_0 = P_r\ \{x=0\} = 0.70$$

$$P_1 = P_r\ \{x=\pm\Delta\} = 0.15$$

באותה דרך בה השתמשנו במודלים הקודמים מקבלים גם את הנוסחאות לאנטרופיה עבור
כל מישור סיבית ℓ ביצוג GRAY של המקור. הנוסחאות הן:

$$H(Y_0) = H(Y_1) = \frac{4}{16} H(P_1) \tag{4.29}$$

$$H(Y_2) = \frac{8}{16} H(P_1) \tag{4.30}$$

$$H(Y_3) = H(P_1) \tag{4.31}$$

ביצוג הבינרי של המקור נקבל:

$$H(Y_0) = \frac{4}{16} H(P_1) \tag{4.32}$$

$$H(Y_1) = \frac{8}{16} H(P_1) \tag{4.33}$$

$$H(Y_2) = H(P_1) \tag{4.34}$$

$$H(Y_3) = H(2P_1) \tag{4.35}$$

כאשר:

$$H(P) \triangleq -P \log P - (1-P) \log (1-P)$$

לאחר הצבת ערכי ההסתברות הופעת הסימבולים P_0 ו-P_1 בנוסחאות נקבל :

$$H_{BINARY}(Y) = 1.9485 \ \frac{\text{סיביות}}{\text{פיקסל}} \text{ ו- } H_{GRAY}(Y) = 1.2197 \ \frac{\text{סיביות}}{\text{פיקסל}}$$

כפי שהתקבל במקרה של א"ב בן 5 ו-7 סימבולים מתקיים גם כאן:

$$H_{GRAY}(Y) < H_{BINARY}(Y)$$

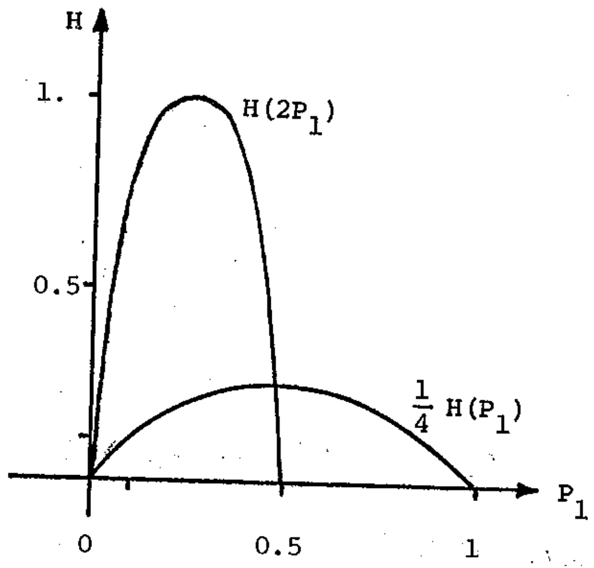
קל יותר לבדוק את תכונותיו של מודל זה משום שנוסחאות האנטרופיה של כל מישור סיבית מורכבות משני ביטויים $H(P_1)$ ו-$H(2P_1)$ בלבד.
לכן הפעם נוכל לבדוק באופן אנליטי האם תמיד עדיף יצוג GRAY של התמונות על פני יצוגן הבינרי. נבדוק עבור אילו ערכי ההסתברות P_1 ו-P_0 מתקיים:

$$H_{BINARY}(Y) - H_{GRAY}(Y) < 0$$

כאשר :

$$H_B(Y) - H_{GRAY}(Y) = H(2P_1) - \frac{1}{4} H(P_1) \tag{4.36}$$

בגרף הבא מופיעים ערכי הביטויים $H(2P_1)$ ו-$\frac{1}{4}H(P_1)$ עבור ערכי P_1.



גרף מס' 4.1: תאור הביטויים $H(2P_1)$ ו-$\frac{1}{4}H(P_1)$.

Graph 4.1: Display for $H(2P_1)$ and $\frac{1}{4}H(P_1)$ expression.

מתוך הגרף ניתן לראות שביטוי (1) מתאפס עבור $P_1 = 0.4792$ (נקודת החיתוך של שתי העקומות). על סמך הקשר בין ההסתברויות של סימבולי המקור X נקבל ש: $P_0 = 1- 2P_1 = 0.0416$. ומכאן ש- $H_{BINARY} < H_{GRAY}$ כאשר:

$0.4792 < P_1 < 0.5$ ו- $P_0 < 0.0416$

השוואת 3 המודלים

בטבלה הבאה מופיעים ערכי "האנטרופיה" $H(Y)$ המתקבלים עבור כל מישור סיבית ב-3 המודלים והערך הממוצע של $\rho(u)$, ומספר הסיביות לפיקסל המתקבל לאחר הדחיסה של 3 תמונות פנים ביצוג GRAY.

<u>טבלה מס' 4.7</u>: ערכי האנטרופיה של המודלים השונים (H(Y)) ומספר הסיביות הממוצע של התמונות האמיתיות((ρ(u)), ביצוג GRAY.

<u>Table 4.7</u>: Entropy values for different models (GRAY representation), H(Y) and ρ(u) (bits/pixel) for real pictures.

מקור	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	H(Y) $\left(\frac{\text{סיביות}}{\text{אות מקור}}\right)$
א"ב 3 אותיות	0.1525	0.1525	0.3049	0.6098	1.2197
א"ב 5 אותיות	0.1727	0.1727	0.3453	0.6343	1.3249
א"ב 7 אותיות	0.1931	0.1931	0.3777	0.6360	1.3999
ממוצע של 3 תמונות (נמדד)	0.2483	0.2767	0.4615	0.6301	1.6166 = ρ(u)

בטבלה הבאה נשווה את ערכי האנטרופיה של המקור X, ואת האנטרופיה הכוללת של 4 מישורי הסיביות, עבור יצוג בינרי ו-GRAY של המקור, של 3 המודלים ליצירת תמונה שתוארו עד כה עם ערך ממוצע של האנטרופיה של תמונות הפרשים H(x) ועם מספר הסיביות לפיקסל המתקבל לאחר דחיסה לפי מימוש 4BP-ZL של האלגוריתם.

<u>טבלה מס' 4.8</u>: ערכי האנטרופיה מסדר אפס ומס' הסיביות לפיקסל ρ(u) המתקבל עבור 3 המודלים ועבור ממוצע של 3 תמונות הפנים.

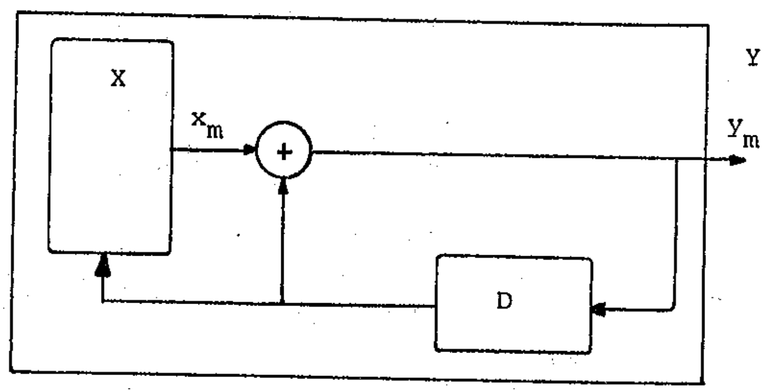
<u>Table 4.8</u>: Zero Order Entropy values and No. bits/pixel ρ(u) for 3 models and average values of 3 face pictures.

מודל	H(x)	(יצוג בינרי) ρ(u)	(יצוג GRAY) ρ(u)	$\left(\frac{\text{סיביות}}{\text{אות מקור}}\right)$
א"ב 3 אותיות	1.1813	1.9475	1.2197	
א"ב 5 אותיות	1.2873	2.0078	1.3249	
א"ב 7 אותיות	1.3548	2.0658	1.3999	
תמונות אמיתיות (ממוצע)	1.3636	2.1454	1.6166	

מתוך התוצאות שהוצגו נובע שהמודל המבוסס על מקור x בעל 7 סימבולים כפי שנותח בסעיף 4.1, הוא המקורב ביותר לערכי הדחיסה שהתקבלו ולערך האנטרופיה H(x) של תמונת הפרשים של תמונות פנים.

4.5.2 <u>מודלים בהם לא משתמשים בסיכום מודולו 256</u>

<u>סכום רגיל:</u> בסיכום רגיל קיימת בעיה של גלישה, דהיינו קבלת ערכי y שאינם שייכים לא"ב Y. כדי למנוע בעיה זו על המקור x לפלוט אותיות המתואמות לערך הקודם של y . סכמת המודל היא:

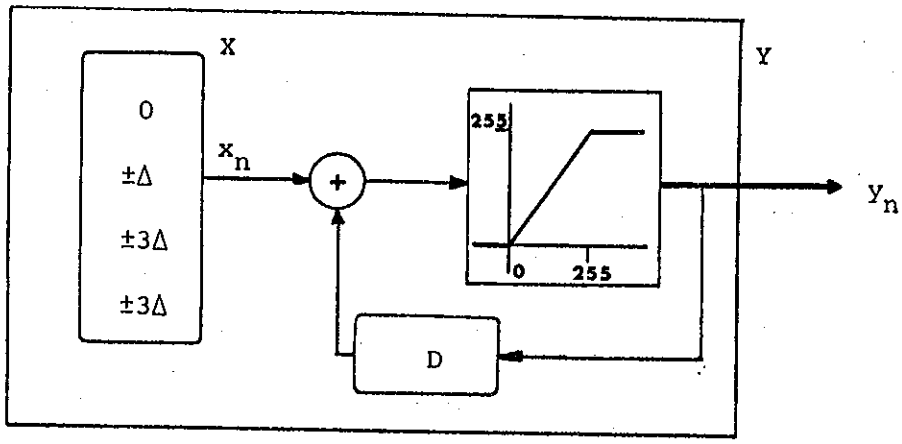


<u>ציור מס' 4.3</u>: מודל ליצירת תמונה המבוסס על סיכום רגיל.
<u>Fig. 4.3</u>: Picture production model with regular adder.

התניה של פליטת סימבול כלשהוא השייכת למקור x בסימבולי המקור y, מגדילה את סיבוכיות המודל, ולכן, הוחלט לא להתיחס למודל זה.

<u>סיכום עם הגבלה (limiting)</u>

מודל יצירת התמונה המבוסס על סיכום עם הגבלה הינו:

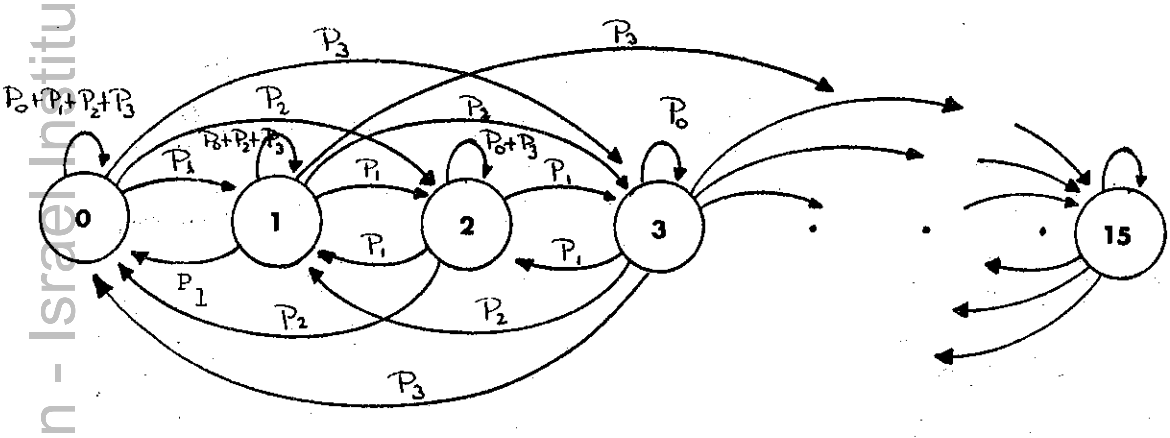


ציור מס' 4.4: מודל ליצירת תמונה המבוסס על סיכום עם הגבלה.

Fig. 4.4: Picture production model with limiter.

מודל זה מכיל מסכם רגיל ו- limiter אשר מבטיח שכל ערכי y_m של התמונה יהיו שייכים לתחום הגדרתם. המקור X במודל זה זהה למקור X אשר הוצג במודל ליצירת התמונה. מקור Y הינו מקור מרקובי מסדר ראשון (להוכחת טענה זו ראה הוכחת טענה מס' 1 בעמ' 70).

דיאגרמת המצבים של שרשרת מרקוב המתארת את המקור Y הינה:



ציור מס' 4.5: דיאגרמת מצבים של שרשרת מרקוב של המקור Y (מודל סיכום עם הגבלה).

Fig. 4.5: State diagram for the Y source Markov Chain. (Model with limiter).

דיאגרמה זו נבדלת מהדיאגרמה המתאימה למודל עם סיכום מודולו 256 בכך שהסתברויות המעברים בין המצבים 2,1,0 למצבים 14,13 ו-15 שוות לאפס, כמו כן ההסתברות להשאר באותו מצב אינה זהה בכל המצבים. במצבים 14,13,2,1,0 ו-15 הסתברות זו גדולה מ-P_0.

נשתמש בסימון $H(P) \triangleq -P\log P-(1-P)\log(1-P)$ כדי להציג את נוסחאות האנטרופיה של כל מישור סיבית של המקור Y. גם במודל זה מתקיים שבמצב המתמיד של שרשרת מרקוב ההסתברות להמצא בכל מצב שווה ל- $\frac{1}{16}$.

עבור ייצוג GRAY של המקור Y נקבל את הביטויים לאנטרופיה של כל מישור סיבית בדרך דומה לזו שבאמצעותה קבלנו את הביטויים עבור המודל עם סיכום מודולו.

ביטויי האנטרופיה הם:

$$H(Y_0) = \frac{2}{16}\left(H(P_3) + H(P_2+P_3) + H(P_1+P_2+P_3)\right) \tag{4.37}$$

$$H(Y_1) = \frac{4}{16}\left(H(P_3) + H(P_2+P_3) + H(P_1+P_2+P_3)\right) \tag{4.38}$$

$$H(Y_2) = \frac{8}{16}H(P_1+P_2+P_3) + \frac{6}{16}H(P_2+2P_3) + \frac{2}{16}H(P_2+P_3) \tag{4.39}$$

$$H(Y_3) = \frac{10}{16}H(P_1+2P_2+P_3) + \frac{2}{16}H(P_1+2P_2) + \frac{2}{16}H(P_1+P_2) + \frac{2}{16}H(P_1+P_2+P_3)$$

$$H(Y) = \sum_{\ell=0}^{3} H(Y_\ell) \tag{4.40}$$

נציב כעת את ערכי הסתברות הופעת הסימבולים של המקור X (P_0,P_1,P_2,P_3) כפי שחושבו בסעיף 4.1.2. בטבלה הבאה נציג את ערכי האנטרופיה של כל מישורי הסיבית המתקבלים במודל זה, את ערכי האנטרופיה של המודל המבוסס על סיכום מודולו ואת מספר הסיביות לפיקסל (ממוצע של 3 תמונות פנים), המתקבל לאחר דחיסת התמונות על ידי מימוש 4BP-ZL של האלגוריתם.

טבלה מס' 4.9: ערכי האנטרופיה של המודלים סיכום מודולו סיכום עם הגדלה וערכי הדחיסה של מימוש 4BP-ZL על תמונות פנים (יצוג GRAY).

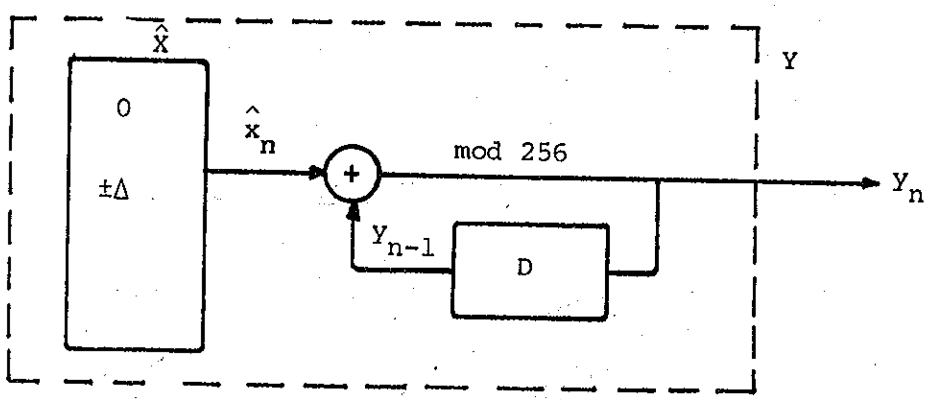
Table 4.9: Entropy values for models: a) Sum Modulo, b) Sum with limiter. Compression values $\rho(u)$ for face pictures.

	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$H(y)$ (סיביות / אות מקור)
מודל סיכום עם הגבלה	0.0965	0.1931	0.3739	0.6268	1.2903
מודל סיכום מודול 256	0.1931	0.1931	0.3777	0.6360	1.3999
תמונות פנים (ממוצע)	0.2483	0.2767	0.4615	0.6301	$1.6166=\rho(u)$

מתוך הערכים שהוצגו בטבלה ניתן להסיק שהמודל המבוסס על סיכום עם הגבלה פחות טוב לתאור תמונות אמיתיות מאשר המודל המבוסס על סיכום מודולו 256. זאת מפני שישנם הפרשים גדולים בין ערכי האנטרופיה של המודל המבוסס על סיכום עם הגבלה לבין תוצאות הדחיסה בסה"כ מישורי הסיבית, ובעיקר במישור הסיבית $\ell = 0$, שבו ההפרש הינו 0.15 סיבית מתוך 0.25 סיבית.

4.5.3 שירשור 2 מקורות מרקוביים מסדר ראשון

סכמת המודל:



ציור 4.6: מודל ליצירת תמונה - שירשור 2 מקורות מרקוביים מסדר ראשון.

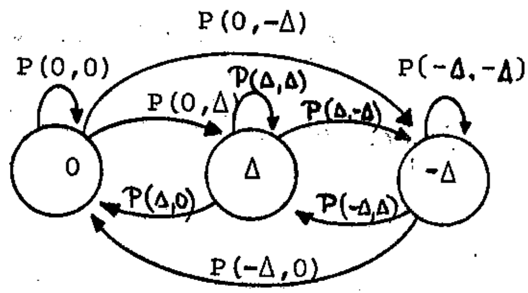
Fig. 4.6: Picture production model - Concatenation of two first order Markov sources.

$$y_n = (y_{n-1} + x_n) \bmod 256 \tag{4.41}$$

המקור $\hat{x}$ הינו מקור מרקובי מסדר ראשון, דהיינו:

$$P_r = \{\hat{x}_n/\hat{x}_{n-1},\hat{x}_{n-2},\ldots,\hat{x}_1\} = P_r\,\{\hat{x}_n/\hat{x}_{n-1}\} \tag{4.42}$$

השרשרת המרקובית המתארת את המקור $\hat{x}$ היא:



ציור מס' 4.7: דיאגרמת המצבים של המקור $\hat{x}$.

Fig. 4.7: State diagram of $\hat{x}$ source.

כדי לקבל תמונה בעלת רצפים ארוכים של אותו סימבול (ביציאה y) יש להעניק הסתברות גבוהה למעברים, בשרשרת מרקוב המתארת את המקור $\hat{x}$, מכל המצבים למצב 0, כלומר למעברים $0 \to 0$, $\Delta \to 0$ ו-$-\Delta \to 0$.

מעברים נוספים בשרשרת שאף הם מבטיחים התנהגות קורלטיבית של ערכי המ"א y_n, אך במידה פחותה, הם המעברים $\Delta \to -\Delta$ ו- $-\Delta \to \Delta$. מטריצת המעבר של שרשרת מרקוב המתארת את המקור $\hat{x}$ הינה:

$$P = \begin{pmatrix} P(0,0) & P(0,\Delta) & P(0,-\Delta) \\ P(\Delta,0) & P(\Delta,\Delta) & P(\Delta,-\Delta) \\ P(-\Delta,0) & P(-\Delta,\Delta) & P(-\Delta,-\Delta) \end{pmatrix} \tag{4.43}$$

כאשר:

$$\sum_{j\in e} P(i,j) = 1$$

לכל i , $i \in e$, $e = \{0,\Delta,-\Delta\}$.

הרחבת המקור $\hat{x}$ מתאפשרת על ידי הוספת סימבולים נוספים ($\pm2\Delta$, $\pm3\Delta$) כפי שנעשה במודל מרקובי מסדר ראשון.

<u>קביעת סדר המקור:</u> נתיחס "לחוג המשוב" במודל, המורכב מיחידת השהייה ומיחידת סיכום, כאל ערוץ ספרתי עם זכרון ונעזר במאפייניו של ערוץ מסוג זה לניתוח המקור Y. התכונות הסטטיסטיות של ערוץ ספרתי עם זכרון מאופיינות [14] על ידי התלות הסטטיסטית שבין הסימבול ביציאת הערוץ (y_n) לבין:

א. הכניסה הנוכחית ($\hat{x}_n$),

ב. סימבולי היציאה ($y_{n-1},y_{n-2},\ldots,y_1$) והכניסה ($\hat{x}_{n-1},\hat{x}_{n-2},\ldots,\hat{x}_1$) הקודמים.

כלומר, ניתן לקבוע את הערך $P(y_1^n/y_1^{n-1},x_1^n)$ כאשר:

$$y_1^n = y_1\, y_2,\ldots,y_n$$

על סמך ביטויים (4.41) ו-(4.42) נקבל עבור המקור Y:

$$P(y_n/y_1^{n-1}\ ,\ \hat{x}_1^n) = P(y_n/y_{n-1},\ \hat{x}_{n-1}) \tag{4.44}$$

זאת אומרת המקור Y המתקבל משירשור 2 מקורות מרקוביים מסדר ראשון הינו מקור מהקובי מסדר ראשון.

מודל זה מתאר בצורה סבירה תמונות בהן קיימים רצפים ארוכים של אותו סימבול, כגון תמונות בעלות אופי Low Pass (המתקבלות כתוצאה מסינון תדרים הגבוהים או Bluring).

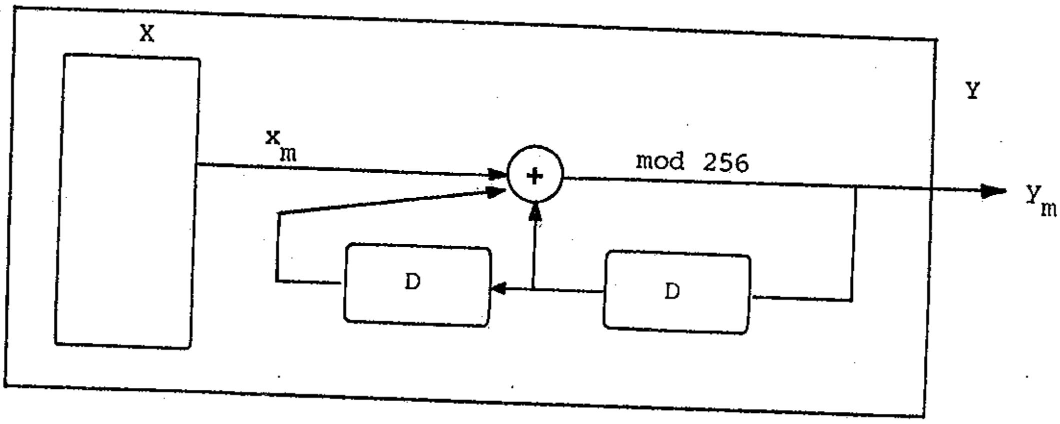
במסגרת עבודה זו מוצג המודל אם כי לא פותח במלואו. בדיקות ראשוניות עם מודל זה לבדיקת רגישותם של מימושים UT ו-4BP-UT הראו שקיימת התאמה טובה בין המודל לתמונות בעלות אופי Low Pass.

4.5.4 מודל מרקובי מסדר שני:

אפשר להרחיב את סדר המודל של השרשרת המרקובית למודל מסדר שני, דהיינו:

$$y_n = (x_n + y_{n-1} + y_{n-2}) \bmod 256$$

סכמת המודל היא:



ציור מס' 4.8. מודל מרקובי מסדר שני ליצירת תמונה.
Fig. 4.8: Picture production model based on a second order Markov source.

כדי לנתח את המודל, יש להמיר מודל מסדר שני בעל 16 מצבים למודל שקול מסדר ראשון בעל $16^2 = 256$ מצבים. המספר הרב של המצבים מעיד על מורכבות הבעיה שבטיפול.

מודל זה אינו מתאר בצורה מספקת את תהליך יצירת התמונה מפני שההשהייה הכפולה (ללא מקדם מתאים) גורמת לכך שערכי y_n הולכים וגדלים כאשר המקור X פולט את הסימבול "0". על ידי כך לא מתאפשרת הווצרות של רצפים ארוכים של אותו סימבול, תכונה הדרושה לתאור תמונות בעלות אופי Low Pass.

פרק 5 : אלגוריתם זיו-למפל בשילוב עם עבוד קדם או אלגוריתמי קידוד אחרים.

עד כה דנו בהפעלת המימושים השונים של אלגוריתם זיו-למפל על תמונות פנים ללא עבוד קדם או שילוב עם שיטות קידוד אחרות. בפרק זה נדון בהפעלת האלגוריתם על תמונה שעברה עבוד קדם או בשילובו עם אלגוריתמי קידוד אחרים במטרה לשפר את מידת הדחיסה.

אחד הקריטריונים העיקריים לבחירת עבודי הקדם ושיטות הקידוד השונות היה הסיבוכיות הנמוכה למימוש; על מנת שהמערכת הכוללת תהיה פשוטה ככל האפשר. לכן, לא נחקרו שיטות המבוססות על התמרת קוסינוס [15] (DCT - Discrete Cosine Transform ועל סכמות קידוד מסתגלות [2] מלבד ADM (Adaptive DM). עבודי הקדם נבחרו גם על פי הקריטריון של אי גרימת עוות נוסף לתמונה, להבדיל מעבודי קדם כמו סינון מעביר נמוכים אשר גורם להפסד פרטים בתמונה. בפרק זה נתאר את עבודי הקדם ושיטות הקידוד שהופעלו ונציג את תוצאות הדחיסה.

5.1 מדדי איכות התמונה

בבדיקת הביצועים של שיטות קידוד תמונות נעזר במספר מדדים מספריים המבטאים באופן חלקי את איכות התמונה המשוחזרת. אנו נבחר במדד SNR שהוא יחס אות לרעש, כאשר "הרעש" הינו שגיאת השיחזור הנגרמת על ידי אלגוריתם הקידוד.

SNR נתון על ידי הביטוי:

$$SNR = 10 \log_{10} \frac{\sigma_u^2}{e^2_{ms}}$$

כאשר σ_u^2 : שונות התמונה המקורית.

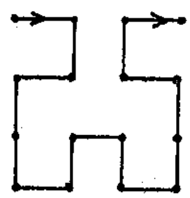
e^2_{ms} : השגיאה הריבועית הממוצעת בין התמונה המקורית לתמונה המשוחזרת.

מדד זה מקובל בספרות המקצועית למרות שאיננו מבטא במהימנות את איכות התמונה המשוחזרת, [6]. בשל כך מקובל להשתמש במדדים סוביקטיביים בהם איכות התמונה נקבעת על ידי צופים. בנספח י' נמצאים תצלומים של התמונות המשוחזרות ושל התמונות המקוריות. "מדידת" איכות התמונה באופן סוביקטיבי ניתנת לביצוע בשתי דרכים. האחת על ידי דירוג של האיכות המוחלטת של התמונה (המשוחזרת) והשניה על ידי השוואת התמונה המשוחזרת עם התמונה המקורית על מנת להבליט את העוותים שהתווספו.

5.2 עבוד קדם ללא גרימת עוות

5.2.1 סריקת תמונות לפי עקומת פאנו (Peano)

לתמונה יצוג דו מימדי. במערכות וידאו הסיגנל המייצג תמונה הינו אות חד מימדי המתאר את עוצמת הבהירות של העצם. המעבר לאות חד מימדי נעשה על ידי סריקה סידרתית של התמונה הנקראת Raster Scan. כאשר יצוג התמונה הוא סיפרתי הסריקה הופכת את המטריצה המיצגת את התמונה לוקטור ארוך המתקבל משירשור שורות המטריצה. הסריקה איננה מתחשבת בתוכן התמונה ולא בתכונות המרחביות שלה, כגון הקורלציה בכוון האנכי. אפשרות אחרת לסריקת התמונה היא על ידי עקומת פאנו (Peano) [16]. פאנו הוכיח ב-1890 שהמסלול של נקודה אחת הנע באופן רציף מעל מרובע יכול, בזמן סופי, לעבור על פני כל נקודה במרובע. עקומת פאנו הבסיסית שעוברת על מטריצה בעלת 4×4 נקודות הינה:



ציור 5.1: עקומת פאנו על 16 נקודות של מטריצה 4×4.

Fig. 5.1: Peano Curve on 16 points of a 4x4 matrix.

סריקת התמונה כולה נעשית על ידי העתקה של המסלול הבסיסי של בלוק 4×4 על כל התמונה לפי מסלול העקומה בבלוק הבסיסי. הסריקה מאחדת נקודות הנמצאות בסביבה מסויימת, לכן, אם נפעיל את אלגוריתם זיו-למפל על סדרת פיקסלים שהתקבלה לאחר סריקת פאנו של תמונה, ניתן לשער שבעזרת האלגוריתם ננצל את הקורלציה הדו-מימדית של התמונה.

למרות ההסברים הנ"ל מתקבלות במציאות תוצאות שונות. ערכי האנטרופיה המותנית $H(y_n/y_{n-1})$, שנמדדת על פיקסלים עוקבים בשורה, גבוהים יותר בתמונות שנסרקו על פי עקומת פאנו מאלה שמתקבלים עבור אותן תמונות ללא סריקת פאנו, כפי שניתן לראות בטבלה הבאה:

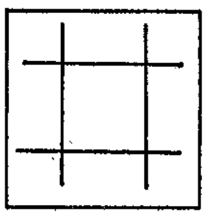
טבלה 5.1: ערכי $H(y_n/y_{n-1})$ ו-$\hat{H}(u)$ המתקבלים מסריקת Raster ומסריקת פאנו של תמונות המיוצגות על ידי 4 סיביות לפיקסל.

Table 5.1: $H(y_n/y_{n-1})$ and $\hat{H}(u)$ values for Raster and Peano scanning of 4 bits pictures.

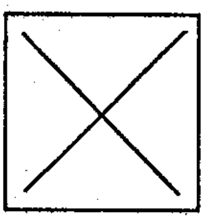
תמונה	$\hat{H}(y_n/y_{n-1})$		$\hat{H}(u)$	
	סריקת Raster	סריקת פאנו	סריקת Raster	סריקת פאנו
1	1.2986	2.6977	1.3854	2.3369
2	1.3382	2.7514	1.8192	2.4694

מתוצאות הטבלה ניתן לראות שגם ערכי שיערוך האנטרופיה $\hat{H}(u)$ גבוהים יותר כאשר התמונה נסרקת לפי עקומת פאנו.

כדי להסביר תופעה זו וכדי לבדוק את השפעת הסריקה על פי עקומת פאנו על ערכי האנטרופיה נבנו שתי תמונות בינריות.



תמונה א'



תמונה ב'

ציור 5.2: תמונות בינריות לבדיקת השפעת הסריקה על פי עקומת פאנו.

Fig. 5.2: Binary pictures for Peano scanning testing.

ערכי האנטרופיה המותנית של שתי תמונות אלה המתקבלים עבור כל אחת משיטות הסריקה מוצגים בטבלה הבאה:

טבלה 5.2: ערכי $H(y_n/y_{n-1})$ עבור תמונות בינריות סינטטיות.

Table 5.2: $H(y_n/y_{n-1})$ values for binary syntetic pictures.

סריקת פאנו	סריקה רגילה	תמונה
0.7556	0.7112	א'
0.4974	0.5399	ב'

מהתוצאות המוצגות בטבלה ניתן להסיק שהסריקה לפי עקומת פאנו "מקלקלת" את הקורלציה בין פיקסלים סמוכים בתמונות בהן רוב הקצוות הם אנכיים ואפקיים, נראה שזהו המקרה בתמונות הפנים שנבדקו.

5.2.2 מפוי הפרשים

מפוי הפרשים הינו התמרה לינארית הנתונה על ידי:

$$y(i,1) = f(1,1)$$

$$y(i,j) = f(i,j-1) - f(i,j) \qquad i = 1,2,\ldots,N \qquad (5.1)$$

$$j = 2,3,\ldots,N$$

כאשר $f(i,j)$ מייצג פיקסל בתמונה המקורית ו-$y(i,j)$ מייצג פיקסל בתמונת ההפרשים.

בעזרת מפוי הפרשים אנו מנצלים את ההשתנות האיטית ברמות האפור בין פיקסלים סמוכים הגורמת לכך שהשונות של ערכי $y(i,j)$ קטנה יותר מזו של ערכי $f(i,j)$. התפלגותם המקורבת של ערכי $y(i,j)$ מתוארת על ידי התפלגות לפלס [6]. יש לציין שעבור מקורות בעלי אותו א"ב האנטרופיה נמוכה יותר במקורות המאופיינים על ידי התפלגות לפלס או גמה מאשר באלה המאופיינים על ידי התפלגות גאוסי או אחידה.

מיפוי הפרשים מגדיל פי 2 את הא"ב הדרוש ליצוג הפיקסלים $f(i,j)$, $i = j = 1,\dots,N$, לכן דרושה סיבית נוספת ליצוג ערכי $y(i,j)$. על מנת להשאר בא"ב המקורי נשתמש במפוי הפרש מודולו 256 (עבור ייצוג ב-8 סיביות של $y(i,j)$) במקום מפוי הפרש רגיל. מכיוון שלא נמצא הבדל בין ערכי האנטרופיה המותנית שהתקבלו מהתמונות הממופות על ידי שתי השיטות למפוי הפרש ("מודולו" ורגיל), עדיף מפוי הפרש "מודולו". בטבלה הבאה מוצגים ערכי $\hat{H}(u)$, $\rho(u)$ ו-$\rho_{4BP}(u)$ עבור תמונות פנים המיוצגות על ידי 4 סיביות לפיקסל לאחר מפוי הפרש מודולו 256. כזכור ערכי 16 רמות האפור נמצאים בתחום 0-255.

טבלה 5.3: ערכי $\hat{H}(u)$, $\rho(u)$ ו-$\rho_{4BP}(u)$ של תמונות פנים המיוצגות על ידי 4 סיביות לפיקסל לאחר מיפוי הפרש מודולו 256.

Table 5.3: $\hat{H}(u)$, $\rho(u)$ and $\rho_{4BP}(u)$ values of difference modulo 256 face pictures (4 bits/pixel).

תמונה	$\hat{H}(u)$	$\rho(u)$	ρ_{4BP}(GRAY)	ρ_{4BP} (בינרי)	(סיביות/פיקסל)
1	1.2263	1.6420	1.6292	2.8727	
2	1.2511	1.6745	1.6433	3.1297	
3	1.0585	1.4226	1.4054	2.7087	
ממוצע	1.1786	1.5795	1.5593	2.9032	
ממוצע ללא מיפוי הפרש	1.3857	1.8180	1.6162	2.1454	

מהתוצאות המוצגות בטבלה עולות הנקודות הבאות:

א. ערכי $\rho(u)$ ו-$\rho_{4BP}(u)$ (יצוג GRAY) המתקבלים מתמונות הפרשים נמוכים מאלה המתקבלים מתמונות מקור (תמונות פנים). ההבדל הממוצע עבור $\rho(u)$ הינו 0.24 סיביות מתוך 1.82 ועבור ρ_{4BP} (יצוג GRAY) 0.06 סיביות מתוך 1.62.

ב. ערכי ρ_{4BP} (יצוג בינרי) של תמונות ההפרשים גבוהים ב-0.75 סיבית מאלה של התמונות הרגילות. ההבדל בערכי ρ_{4BP} בין תמונות ההפרשים לתמונות הרגילות תלוי ביצוג התמונה (בינרי או GRAY), וזאת משום שככל שהתפלגות המ"א פחות אחידה ההפרש בין תוצאות שני היצוגים גדול יותר.

השיפור בערכי $\rho(u)$ ו-$\rho_{4BP}(u)$ (ביצוג GRAY) איננו משמעותי מכאן נובע שמפוי הפרשים לא משפיע באופן ניכר על יכולת הדחיסה של אלגוריתם זיו-למפל. יש לציין גם שמפוי הפרשים פגיע יותר לשגיאות ערוץ מכיוון ששגיאת ערוץ אחת נגררת לאורך כל תהליך הקידוד.

5.3 קידוד עם עוותים

במטרה לשפר את תוצאות הדחיסה נדון בסעיף זה בשילוב של אלגוריתם זיו-למפל עם שיטות קידוד מקובלות לדחיסת תמונות הגורמות לעוות בתמונה המקודדת. הפעלת אלגוריתם ZL נעשית על מילות הקוד המתקבלות משיטות קידוד אלה. קיים מספר רב של שיטות קידוד המופעלות על סמך עקרונות שונים [3], [4]. מתוכן נדון בטכניקות קידוד שסיבוכיות מימושן נמוכה וזאת כדי שהמערכת המבצעת את הקידוד תהיה פשוטה והזמן הדרוש לקידוד ולפיענוח יהיה קצר ככל האפשר.

5.3.1 קידוד אינטרפולטיבי

בשיטה זו משדרים רק חלק מהפיקסלים של התמונה המקורית, והפיקסלים שלא שודרו משוחזרים על ידי אינטרפולציה או על ידי רפליקציה. קיימות מספר דרכים לקידוד אינטרפולטיבי, ביניהן דצימציה מרחבית שבה קצב הדגימה של התמונה נמוך יותר מהקצב המקורי. מכיוון שלא נשתמש במסננת מעבירת נמוכים (כדי למנוע את הצורך בכח חישובי רב לביצוע הסינון) נקבל קיפול בתדר המרחבי (aliasing) וכמו כן רזולוצית התמונה תהיה פחות טובה.

<u>דצימציה מרחבית 2:1</u> מתבצעת על ידי דילוג על כל פיקסל שני (או על כל שורה שניה). נסמן ב-$f(m,n)$ את ערך הפיקסל בתמונה המקורית וב-$y(m,n)$ את ערך הפיקסל של התמונה לאחר הדצימציה. מתקיים אז הקשר הבא:

$$y(m,n) = f(2m,n) \qquad n = 0,1,\dots,N-1 \qquad (5.2)$$
$$m = 0,1,\dots,\frac{N}{2} - 1$$

כאשר N: מספר הפיקסלים בשורה (או בעמודה) של התמונה המקורית.

לאחר הדצימציה מספר הפיקסלים בתמונה הינו $N^2/2$. שיחזור הפיקסלים החסרים נעשה על ידי רפליקציה (שוב, כדי לחסוך בחישובים), זאת אומרת על ידי העתקה של ערך הפיקסל הקודם.

בטבלה הבאה מופיעים ערכי הדחיסה $\hat{H}(u)$ (משערך לאנטרופיה), $\rho(u)$, $\rho_{4BP}(u)$ (יצוג GRAY) ו-$\rho_{4BP-UT}(u)$ (פיסוק בעזרת עץ אוניברסלי סינטטי) וערכי SNR עבור תמונות פנים המיוצגות על ידי 4 סיביות לפיקסל.

טבלה 5.4: ערכי דחיסה (סיבית לפיקסל) ו-SNR עבור תמונות המיוצגות על ידי 4 סיביות לפיקסל שעברו דצימציה מרחבית 2:1.

Table 5.4: Compression values and SNR of 4 bits/pixel face pictures after 2:1 spatial decimation.

תמונה	$\hat{H}(u)$	$\rho(u)$	$\rho_{4BP}(u)$ (*)	$\rho_{4BP-UT}(u)$ (*)	SNR [dB]
1	0.708	0.964	0.869	0.899	12.82
2	0.780	1.059	0.862	0.890	26.14
3	0.641	0.875	0.742	0.761	28.90
ממוצע	0.710	0.966	0.825	0.850	22.62
ממוצע עבור תמונות מקוריות	1.385	1.818	1.617	1.701	

(*) יצוג GRAY.

השיפור בדחיסה המתקבל על ידי דצימציה מרחבית 2:1 הוא כמעט פי 2. איכות התמונה המשוחזרת (ראה תמונות בנספח י') טובה, פרט לשיחזור האותיות המופיעות בכותרת התמונה.

<u>דצימציה מרחבית 4:1</u> התבצעה על ידי דילוג על כל שורה ועל כל עמודה, כאשר מתקיים הקשר הבא בין הפיקסלים של התמונה המקורית והתמונה המקודדת:

$$y(m,n) = f(2m,2n) \qquad m = n = 0,1,\ldots,\frac{N}{2}-1 \tag{5.3}$$

לאחר הדצימציה מס' הפיקסלים בתמונה $y(m,n)$ הינו $N^2/4$. שוב כמו במקרה הקודם שיחזור הפיקסלים החסרים נעשה על ידי רפליקציה.

בטבלה הבאה מופיעים ערכי הדחיסה $\hat{H}(u)$, $\rho(u)$ ו-$\rho_{4BP}(u)$ וערכי SNR עבור תמונות פנים המיוצגות על ידי 4 סיביות לפיקסל.

<u>טבלה 5.5</u>: ערכי דחיסה ו-SNR עבור תמונות פנים המיוצגות על ידי 4 סיביות לפיקסל שעברו דצימציה מרחבית 4:1.

<u>Table 5.5</u>: Compression values and SNR of 4 bits/pixel face pictures after 4:1 spatial decimation.

תמונה	$\hat{H}(u)$	$\rho(u)$	$\rho_{4BP}(u)$ $(\frac{\text{סיביות}}{\text{פיקסל}})^{(*)}$	SNR [dB]
1	0.432	0.578	0.581	10.63
2	0.446	0.596	0.530	20.14
3	0.369	0.500	0.459	20.95
ממוצע	0.416	0.558	0.523	17.24
ממוצע עבור תמונות מקוריות	1.358	1.818	1.617	

(*) יצוג GRAY.

מתוצאות הטבלה נובע שהשיפור בדחיסה המתקבל על ידי קידוד של רבע מכמות הפיקסלים (דצימציה 4:1) בתמונה הינו פי 3 ויחס הדחיסה הכולל במקרה זה הוא $7.6:1 = (\frac{4}{0.523})$.

איכות התמונה המשוחזרת נמוכה, מופעים ארטיפקטים כגון שן משור בקצוות העצם, ולא ניתן לשחזר אותיות.

איכות התמונה המשוחזרת על ידי אינטרפולציה לינארית אף היא אינה טובה, מכיון שנוצר מיצוע, הגורם לטישטוש הקצוות (Bluring).

5.3.2 קידוד פרדיקטיבי (Predictive Coding)

שיטת קידוד זו נועדה להסיר את היתרות הקיימת בין פיקסלים סמוכים על ידי חיזוי וכימוי (quantize) של המידע החדש, כלומר שגיאת החיזוי. בין טכניקות הקידוד הפרדיקטיביות המקובלות, השיטות הידועות הן: DPCM, ADPCM, DM ו-ADM.

קידוד דלטה מסתגל (ADM)

קידוד דלטה (DM) היא הטכניקה הפשוטה בין שיטות הקידוד הפרדיקטיבי. מגבלותיה העיקריות הן: Slope overload ו-Granularity noise, תופעות המשפיעות על איכות התמונה. המגבלה הראשונה משפיעה על מיקום קצוות העצם בתמונה והשניה מוסיפה רעש לאיזורים בתמונה בהם רמת האפור לא משתנה. אלו הן שתי הפרעות שהעין מאוד רגישה אליהן.

על מנת להתגבר על חסרונות אלה, וכתוצאה מכך לשפר את ביצועי המקדד, נשתמש בסכמה מסתגלת (ADM) המבוססת על האלגוריתם של SONG [17]. המשוואות המגדירות את המקדד הן:

$$
\begin{aligned}
E_k &= \operatorname{sgn}(S_k - X_k) \\
X_k &= X_{k-1} + \Delta_k \\
\Delta_k &= \begin{cases} |\Delta_{k-1}| \, (E_{k-1} + \frac{1}{2} E_{k-2}) & |\Delta_{k-1}| \geq 2\Delta \\ 2|\Delta| \quad E_{k-1} & |\Delta_{k-1}| < 2\Delta \end{cases}
\end{aligned}
\tag{5.4}
$$

כאשר: S_k : הכניסה למקדד.

E_k : היציאה של המקדד (סיבית אחת).

X_k : שיערוך של אות הכניסה ברגע k .

Δ_k : גודל צעד הקוונטיזציה.

Δ : קבוע אשר קובע את הערך המינימלי של Δ_k.

בסכמה זו מופקת סיבית אחת עבור כל פיקסל. קצב ההשתנות של צעד הקואנטיזציה (בתלות באות) נמצא בהתאמה עם ההשתנות האופטימלית הדרושה עבור אותות וידאו כפי שנקבע במחקריהם של Song ו-Cutler [17].

בבדיקות שבוצעו על תמונות פנים המיוצגות על ידי 8 סיביות לפיקסל התקבלו ערכי $\rho(u)$ הבאים:

טבלה 5.6: ערכי $\rho(u)$ ו-SNR עבור תמונות פנים המיוצגות על ידי 8 סיביות לפיקסל לאחר קידוד ADM.

Table 5.6: $\rho(u)$ values and SNR values of ADM coded face pictures.

תמונה	1	2	3	ממוצע	
$\rho(u)$	0.752	0.750	0.743	0.748	סיביות לפיקסל
SNR	10.85	14.99	16.30	14.05	[dB]

מתוצאות הטבלה נובע שיחס הדחיסה הכולל שהושג הינו 10.6:1. השיפור בדחיסה משמעותי אך איכות התמונה המקודדת בדרך זאת היא נמוכה מכיוון שגבולות העצמים מטושטשים ואין אפשרות לשחזר אותיות.

5.3.3 קידוד מקדמי התמרה (Transform Coding)

קידוד מקדמי התמרות הוכח כדרך יעילה לקידוד תמונות [8], [3]. על פי הגישה הבסיסית לקידוד תמונות בעזרת התמרות, מחלקים את התמונה לבלוקים קטנים של פיקסלים, ומבצעים התמרה דו-מימדית על כל בלוק לחוד, כדי לקבל מערך מקדמים בעל אותו מימד. לאחר מכן יש לבצע כימוי (קוונטיזציה) וקידוד על המקדמים לצורך שידורם או איחסונם.

בקידוד מקדמי התמרה, הדחיסה מושגת גם על ידי איפוס של מקדמים בעלי שונות קטנה וגם על ידי קוונטיזציה גסה של המקדמים הנותרים הנקבעים על פי האיכות שנדרשת מהתמונה המשוחזרת. הפרמטרים המשפיעים על ביצועי הקידוד בעזרת התמרה הינם: גודל הבלוק, סוג ההתמרה, בחירת מקדמים שיאופסו, שיטת הכימוי של המקדמים, והקצבת מילות קוד למקדמים. על מנת שאיפוס המקדמים יגרום

לשגיאת שיחזור מינימלית דרוש שההתמרה תרכז במספר מועט של מקדמים את רוב האנרגיה של בלוק הפיקסלים. ההתמרה האופטימלית מבחינה זו הינה התמרת KL Karhunen-Loeve, אך מימושה מסורבל וקשה לביצוע [4]. קיימות התמרות נוספות (תת-אופטימליות) מביניהן הפשוטה ביותר למימוש היא התמרת הדמרד (Hadamard Transform) [19].

1. התמרת הדמרד (Hadamard Transform)

בסעיף זה נבדוק שילוב של קידוד תמונות על ידי התמרת הדמרד הדו-מימדית עם דחיסת המקדמים לאחר הכימוי בעזרת אלגוריתם זיו-למפל. התמרה זו נבחרה בשל הסיבוכיות הנמוכה למימושה ומכיוון שקיים אלגוריתם מהיר (Fast Hadamard Transform) לביצועה (Pratt et al.) [19]. ההתמרה הדו-מימדית מחושבת על ידי חישוב ההתמרה החד-מימדית על שורות התמונה ולאחר מכן על עמודות התמונה הנוצרת מהשורות המותמרות. חישוב ההתמרה דורש פעולות סיכום וחיסור בלבד. מספר הפעולות הדרוש לחישוב התמרה חד מימדית של וקטור בעל N איברים הינו N log N (חישוב התמרה בדרך הישירה דורש N^2 פעולות). חישוב ההתמרה ההפוכה נעשה בדרך זהה לזו של ההתמרה הרגילה, פרט למקדם הנירמול.

תאור פורמלי של התמרת הדמרד: קיימות 2 דרכים מקובלות לחישוב התמרת הדמרד. באחת מחשבים את המקדמים לפי הסדר הטבעי (לפי ההגדרה המקורית של ההתמרה) ובשניה מחשבים את המקדמים לפי סדר עולה של תדרי פונקציות הבסיס. אלגוריתם Pratt מתבסס על הדרך השניה.

נסמן ב-f(x,y) פיקסל של תמונה מקורית. ניתן לרשום את התמרת הדמרד הדו-מימדית על ידי:

$$F(u,r) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\ (-1)^{q(x,y,u,v)} \tag{5.5}$$

כאשר:

$$q(x,y,u,v) \triangleq \sum_{i=0}^{n-1} [g_i(u)\ x_i + g_i(v)\ g_i]$$

האיברים x_i ו-y_i הם היצוג הבינרי של x ו- y בהתאמה, לדוגמא:

$$(x)_{\text{decimal}} = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)_{\text{binary}}$$

וכן:

$$g_0(u) = u_{n-1} \; ; \quad g_1(u) = u_{n-1} + u_{n-2} \; ;$$

$$g_{n-1}(u) = u_1 + u_0$$

כאשר u_i ו-v_i הינן **הסיביות** ה-i-יות של היצוג הבינרי של u ו-v בהתאמה. מעניין לציין שפונקצית $g_i(u)$, $i = 0,1,\ldots,n$ מתארות את היצוג לפי קוד GRAY של u .

<u>ביצוע ההתמרה.</u>

חישוב ההתמרה נעשה בעזרת האלגוריתם המהיר של Pratt. הפעלתו על התמונה נעשית על פי העקרון המוצג במאמר של Landau & Slepian [20]. העיקרון מבוסס על חישוב התמרת הדמרד החד מימדית על וקטור בעל מימד m^2 המתקבל על ידי שירשור שורות בלוק בגודל mxm. לפיכך השלבים הם:

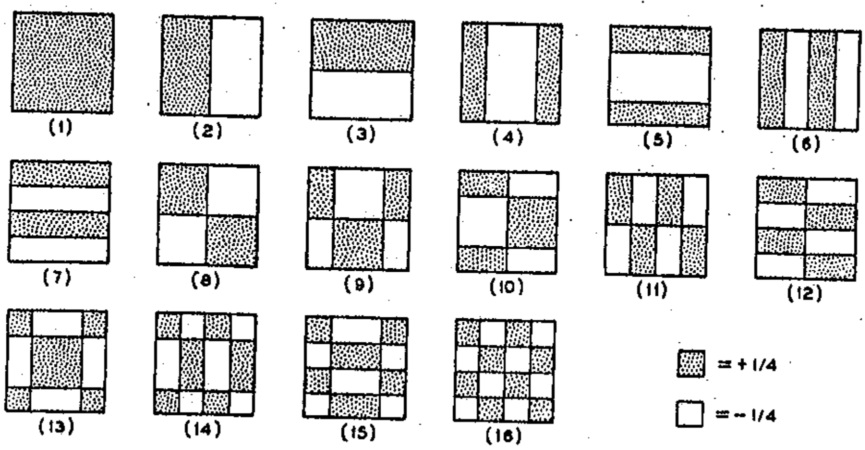
1) חלוקת התמונה בגודל 256×256 לבלוקים בגודל mxm ($m = 4$);

2) שירשור m שורות הבלוק כדי לקבל וקטור Y, כלומר:

$$Y = [f_{11} \; f_{12}, \ldots, f_{1m} \; \ldots \; f_{mm}]^T$$

כאשר עבור $m = 4$ מימד הוקטור הוא 16.

דרך זו אקויולנטית לביצוע התמרה דו-מימדית על המטריצה המקורית. בעבודתם של Landau & Slepian חישוב התמרה נעשה על ידי 16 פונקציות בסיס שייצוגן הדו-מימדי (ראה ציור 5.3) הוא בעל תדר מרחבי עולה עם $b_i : i$, $i = 1,2,\ldots,16$, כאשר: $Y = \sum_{i=1}^{16} c_i b_i$ ו- c_i הם מקדמי ההתמרה.



ציור 5.3: פונקציות בסיס הדמרד (מתוך [20]).

Fig. 5.3: Hadamard basis functions, from [20].

חישוב ההתמרה לפי האלגוריתם של Pratt נעשה בעזרת אותן פונקציות בסיס אך סדורן שונה, כלומר נקבל מקדמי התמרה a_i, $i = 1,2,\ldots,16$ בסדר i עולה, המתאימים למקדמים c_i אך בסדר הבא:

$$c_1, c_3, c_5, c_7, c_{12}, c_{10}, c_8, c_2, c_4, c_9, c_{13}, c_{15}, c_{16}, c_{14}, c_{11}, c_6$$

כימוי המקדמים

שלב זה הינו חשוב ביותר, כיוון שעל ידי בחירת הכימוי המתאים של המקדמים נוכל להפיק תוצאות דחיסה טובות מבלי לגרום לעוותים שישפיעו על איכות התמונה. על מנת שסיבוכיות המערכת תהיה נמוכה, נשתמש בטכניקות פשוטות שאינן כוללות שיטות מסתגלות של כימוי. לכן, נקבע מראש, על סמך שונות המקדמים של קבוצת תמונות מסויימת אילו מהמקדמים יאופסו ומהו מספר הסיביות שיוקצבו למקדמים האחרים. שיטה זו נקראת Zonal Sampling (3).

מספר הסיביות האופטימלי לכל מקדם המאפשר קבלת שגיאת שיחזור מינימלית (mse) נתון על ידי הביטוי [22]:

$$R_i = R + \frac{1}{2} \log \frac{\sigma_i^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2\right]^{1/N}} \quad \frac{\text{סיביות}}{\text{דגימה}} \qquad (5.7)$$

$$i = 1,2,\ldots,16$$

כאשר: R_i: מס' הסיביות עבור המקדם ה-i.

R: מספר הסיביות הרצוי לכל פיקסל: $R \triangleq \sum_{i=1}^{N} \frac{R_i}{N}$

N: מספר המקדמים בהתמרה.

σ_i^2: שונות המקדם ה-i.

על פי נוסחה (5.7) נקבל שעבור קבוצת תמונות פנים מספרי הסיביות למקדמים C_i הינם $(R = 2)$:

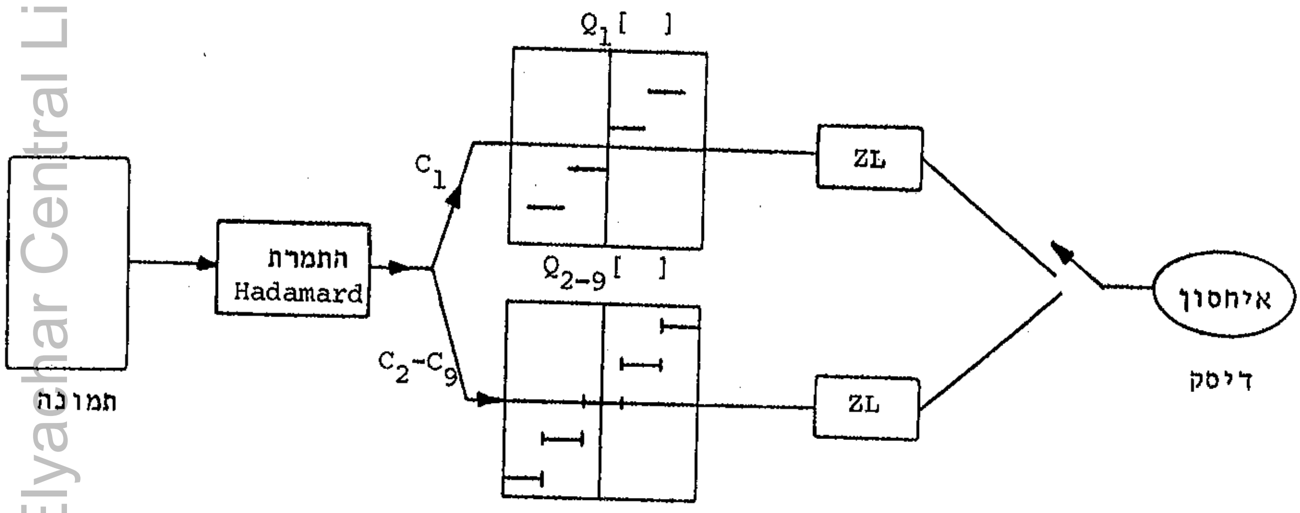
C(i) :	1	2	3	4	5	6	7	8	9	10	11-16
R_i :	6	4	4	3	3	3	3	2	2	2	0

תוצאות אלה מתאימות להקצבה עליה הוחלט בעבודתם של Slepian ו-Landau. בעבודה זו הם משתמשים במכמה שונה עבור כל קבוצת מקדמים שהוחלט להקציב להם מס' זהה של סיביות. לשם כך תוכננו 4 מכמים לא אחידים (Non Uniform) שונים, פרט למכמה של C_1 שהוא אחיד.

בעבודתנו הוחלט להשתמש בשני מכמים אחידים כדי לפשט את תהליך הכימוי. עבור המקדם C_1 המכמה הוא זוגי (Midrise Quantizer) בעל $2^6 = 64$ רמות יציאה. המכמה השני, עבור המקדמים C_2 עד C_9, הינו אי זוגי (Midtread Quantizer) בעל 15 רמות יציאה (4 סיביות). משתמשים במכמה אי זוגי עבור המקדמים C_9-C_2 מכיוון שהתפלגותם מקורבת להתפלגות לפלס [21], וניתן להשיג אנטרופיה נמוכה יותר ביציאתו של מכמה אי זוגי מאשר בזו של מכמה זוגי [22]. מקדמים C_{10} עד C_{16} לא משתתפים בהתמרה ההפוכה ומניחים שערכם אפס.

תחום ערכי המקדמים הינו: מקדם C_1 $[0 \div 255]$, מקדמים אחרים $[-127 \div 127]$.

צעד הכימוי Δ במכמה של C_1 הינו 4 והוא נקבע על ידי חלוקה של תחום הערכים של C_1 (0-255) במספר רמות היציאה של המכמה (64). במכמה השני תחום היציאה וצעד הכימוי Δ נקבע על פי חישוב השגיאה הריבועית המינימלית. התקבל $\Delta = 2$ אך איכות התמונה המשוחזרת לא השביעה רצון, לכן הוחלט לקבוע את גודל הצעד על פי איכות התמונה המשוחזרת בה נמצא כי $\Delta = 4$ ואז תחום ערכי היציאה של המכמה הוא [+28, -28], מפני שבו ישנן 15 רמות יציאה כאשר הרמה האמצעית היא רמת 0. סכמת הקידוד המשולבת של התמרת הדמרד ואלגוריתם ZL מופיעה בציור הבא:



ציור 5.4: קידוד בעזרת התמרת הדמרד ואלגוריתם אוניברסלי של זיו-למפל.

Fig. 5.4: Coding by Hadamard Transform and the ZL algorithm.

לאחר ביצוע התמרה על תמונה המיוצגת על ידי 8 סיביות לפיקסל וכימוי של המקדמים מתקבל שמס' הסיביות לפיקסל הדרוש ליצוג התמונה הינו $R = 2.375$. מספר זה מתקבל על ידי חלוקה של סה"כ מספר הסיביות הדרושות להצגת המקדמים (38) במספר המקדמים (16). תוצאות הדחיסה על פי המימושים השונים של אלגוריתם זיו-למפל מופיעות בטבלה הבאה:

טבלה 5.7: ערכי הדחיסה ו-SNR המתקבלים מקידוד תמונות המיוצגות על ידי 8 סיביות לפיקסל בעזרת התמרת הדמרד ואלגוריתם ZL.

Table 5.7: Compression values and SNR of 8 bits/pixel codes pictures by Hadamard Transform and the ZL algorithm.

תמונה	$\hat{H}(u)$	$\rho(u)$	$\rho_{4BP}(u)$ (1)	$\rho_{4BP-UT}(u)$ (1)(2)	SNR [dB]
1	0.783	1.093	1.099	1.058	16.28
2	0.775	1.080	1.021	0.996	22.57
3	0.731	1.022	0.944	0.992	23.87
ממוצע	0.763	1.065	1.022	1.015	20.91

(1) יצוג GRAY

(2) עצי הפיסוק נבנו בעזרת מקדמי הדמרד שהתקבלו מתמונות פנים אחרות.

השתמשנו בעץ יחיד השייך למישור הסיבית - MSB, לכל ששת מישורי הסיבית של המקדם C_1 ובעצים שונים עבור כל אחד ממישורי הסיבית של המקדמים C_9-C_2. סה"כ חמישה עצים.

מתוצאות הטבלה נובע שיחס הדחיסה הכולל שהושג הוא כ-8:1 ויחס הדחיסה עבור המקדמים בלבד באמצעות אלגוריתם ZL הוא כ-2.3:1. הושגו כך תוצאות דחיסה טובות בהחלט באיכות תמונה משוחזרת שהיא טובה מאוד.

בעזרת טכניקת קידוד זו ניתן להשיג תוצאות טובות מאלה שמוצגות בטבלה. משום שמספר המקדמים שקודדו אינו מספיק בכדי שהאלגוריתם ילמד את התכונות הסטטיסטיות של המקדמים, ובשל כך ניתן להבין מדוע ערכי ρ_{4BP-UT} שהתקבלו עבור עץ יחיד לכל הרמות (במקרה של C_1) נמוכים אך במעט מערכי ρ_{4BP}. על מנת להתגבר על בעיה זו רצוי לקדד רצף של מספר תמונות.

אפשרות אחרת לשיפור המערכת היא לפתח מודלים מתאימים שיאפשרו שימוש בעצים סינטטיים לקידוד המקדמים.

בדיקת אפשרויות אלה לשיפור המערכת יכולה להוות נושא להמשך המחקר.

פרק 6 : סכום ומסקנות

בעבודה זו נבדקו ביצועי אלגוריתם זיו-למפל על תמונות פנים. מצאנו כי תוצאות הדחיסה המתקבלות מהפעלת האלגוריתם על תמונות המיוצגות על ידי 8 סיביות לפיקסל (דחיסה ללא עוות), אינן מספקות. האלגוריתם הופעל לכן על תמונות פנים המיוצגות על ידי 4 הסיביות המשמעותיות ביותר. איכות תמונות אלה טובה בהחלט.

מניתוח ומבדיקת המימושים היסודיים של האלגוריתם המבוססים על עצי פיסוק עולה שלמרות שהדחיסה בעזרת המימוש אשר מתחשב בגבולות ה-Byte אינה עונה על כל הדרישות, היא טובה יותר מזו המתקבלת בעזרת המימושים המבוססים על עץ מדרגה α ועל עץ בינרי רגיל. פיסוק על ידי עץ בינרי רגיל אינו מנצל מספיק את הקורלציה בין פיקסלים סמוכים. הפיסוק על ידי עץ מדרגה α אינו יעיל מפני שרק אחוז קטן ממילות הקוד הקיימות בעץ מנוצל.

הצעתי לכן, בעבודה זו, מספר מימושים חדשים המבוססים על עצי פיסוק בינריים. במימוש הקרוי 4BP, קידוד התמונה נעשה על כל אחד מ-4 מישורי הסיבית בנפרד. על כן ניתן לקבל תוצאות דחיסה טובות (יחס דחיסה של כ-2.5:1) כאשר המימוש מופעל על תמונות המיוצגות לפי קוד GRAY.

הסיבה העיקרית לביצועיו הטובים נובעת מניצול הקורלציה הגבוהה שקיימת בין סיביות עוקבות השייכות ל-2 מישורי הסיביות המשמעותיות ביותר. קורלציה זו גבוהה מזו הקיימת בין פיקסלים סמוכים. יתרון נוסף של המימוש הוא היכולת לשיחזור הדרגתי של תמונות המאיצה את החיפוש במאגרי הנתונים.

בעזרת מימוש UT-ZL (עץ פיסוק אוניברסלי) ניתן לקבל מילות קוד באורך קבוע ולשפר את החסינות בפני רעש. אלה הם יתרונות חשובים כאשר מישמים את האלגוריתם למטרות תקשורת. אך, מימוש זה המבוסס על עץ פיסוק בינרי תוך שמירת גבולות ה-Byte, רגיש למבנה הסטטיסטי של התמונה, רגישות הפוגעת ביכולת דחיסתו.

במימוש 4BP-UT-ZL משולבים שני העקרונות של המימושים הקודמים. בנוסף ליתרונות של העץ האוניברסלי, גם תוצאות הדחיסה אינן נפגעות. הסיבה העיקרית לאי הרגישות של מקדם הדחיסה למבנה הסטטיסטי של התמונה נובעת מכך שעץ פיסוק של סימבולים בינריים מכיל יותר מידע על התכונות הסטטיסטיות של המקור (מישורי

סיבית) מאשר עץ פיסוק של סימבולים השייכים לאלפבית המקורי של התמונה כאשר ל-2 העצים אותה כמות זכרון. באמצעות מודל ליצירת תמונה שפותח בעבודה זו התקבלה הערכה טובה של ביצועי המימוש 4BP.

שימוש בעצים סינטטיים שנבנו בעזרת המודל מאפשר חסכון בזכרון הדרוש לאיחסונם וקבלת תוצאות דחיסה טובות, אם כי בתהליך הפיענוח של התמונות המקודדות יש לבנות את העצים מחדש. בנוסף, יש למדוד בתמונות את הפרמטרים הדרושים למודל.

על מנת לשפר את תוצאות הדחיסה נבדק שילוב האלגוריתם עם טכניקות קידוד אחרות. מבין הטכניקות שנבדקו ראויה לציון התמרת הדמרד (Hadamard) שבאמצעותה התקבל יחס דחיסה של כ-8:1 ואיכות תמונה טובה מאוד. יחד הדחיסה המושג בעזרת כימוי מקדמי ההתמרה הוא 8:2.375, ויחס הדחיסה המתקבל על ידי קידוד מקדמי ההתמרה בעזרת מימוש BP-UT-ZL הוא 2.375:1.

המסקנה הניתנת להפקה מעבודה זו היא שלמימושים החדשים של אלגוריתם זיו-למפל יישום יעיל לקידוד תמונות עבור מאגרי נתונים. באמצעותם מתקבלים תהליכי קידוד ופיענוח בעלי סיבוכיות נמוכה, תוצאות דחיסה טובות וזמן קבוע (ללא תלות באורך המחרוזת) לפיענוח מילות קוד.

אחד הכוונים האפשריים להמשך המחקר הינו שילוב האלגוריתם עם טכניקות קידוד נוספות והתאמת המימושים לטכניקות אלה.

כוון אחר הינו עידון המודל הקיים ופיתוח מודלים אחרים על מנת ליצור מימושים נוספים המתאימים לקידוד התמונות.

ניתן גם להמשיך את המחקר על ידי השלמת הפרטים הטכניים של המימוש 4BP-UT-ZL כגון, הוספת פולסי סינכרון, ואמצעים אחרים להבטחת חסינות גבוהה לשגיאות ערוץ, וחוצצים לקבלת קצב קבוע של פליטת מילות קוד שיאפשרו ישום המימוש למטרות תקשורת.

נספח א': התמונה כמקור אינפורמציה

מקורות אינפורמציה מוגדרים על ידי:

1) אלפבית של המקור U, $U \in R$;

2) וקטור של אותיות מקור $\underline{u}$, $\underline{u} = (u_1, u_2, \ldots, u_N)$, $\underline{u} \in U^N \subset R^N$;

3) איפיון הסתברותי של המקור: פילוג של אותיות המקור, השתנותו של הפילוג בזמן והקורלציה בין הסימבולים שהמקור פולט;

4) קצב המקור: ρ_s אותות לשניה;

5) פונקצית העוות של המקור $d(\underline{u},\underline{v})$, $\underline{u},\underline{v} \in U^N$;

על סמך חמישה פרמטרים אלה אפשר להגדיר תמונה כיצוג סיפרתי כמקור אינפורמציה בעל א"ב של 256 אותיות $U = \{0,1,\ldots,255\}$, כאשר משתמשים ב-8 סיביות עבור אלמנט תמונה (א"ת) הנקרא "פיקסל" ("pixel": Picture element).

במקרה זה '0' מסמן צבע שחור ו-255 מסמן צבע לבן.

$\underline{u}$ הוא וקטור במימד n כאשר n יכול לקבל את הערך $n = 1$ (פיקסל יחיד) או את הערך של מספר הפיקסלים בשורה אחת או הערך של מספר הפיקסלים המתקבל משירשור של מספר שורות של תמונה אחת או אף של מספר תמונות. כמו כן ניתן להתיחס לתמונה כאל שדה אקראי [7] אם לוקחים בחשבון את דו-המימדיות של התמונה. במחקר זה הוקטור $\underline{u}$ מקבל מימד n כלשהוא שתלוי במקרה הנדון.

איפיון הסתברותי של תמונה כפי שמתואר בספרות [4], מצביע על תלות בין פיקסלים סמוכים וכמו כן על אי-סטציונריות. במודלים שאנו נטפל נתיחס לתמונה כאל מקור מרקובי מסדר ראשון ונתבונן בגודל $P_n(u_i/u_j) \triangleq$ הסתברות שהמקור יפלוט ברגע n את הסימבול u_i כאשר ברגע $n-1$ פלט את הסימבול u_j. לצורך פישוט בעיות תאורטיות נניח סטציונריות של התמונה, למרות שהנחה זאת איננה מדויקת באופן כללי, אך היא מתאימה לאיזורים בהם הפעילות נמוכה (רמות אפור משתנות לאט).

קצב המקור (במובן של מספר הסימבולים לשניה) נקבע על ידי המערכת בה נמצאת התמונה (למשל: מחשב, יחידת דיסק, מערכות מיוחדות לעיבוד תמונות) העוברת קידוד או על ידי ערוץ התקשורת שדרכו מועברת התמונה.

פונקצית העוות המקובלת לבדיקת תוצאות הדחיסה היא פונקצית השגיאה הריבועית הממוצעת (m.s.e.) אף על פי שמדד זה איננו מבטא את השפעת שגיאות השיחזור השונות שהעין רגישה להם. לכן בדיקת איכות השיחזור נעשית גם על ידי השוואת התוצאות על ידי צופה, דהיינו מדד סוביקטיבי. בעבודה זו נציג את שני המדדים הנ"ל, הראשון על ידי יחס אות לרעש (SNR) כאשר לשגיאת הקוונטיזציה מתייחסים כרעש והשני על ידי הצגת התוצאות בעזרת תצלומים.

נספח ב': מדידת האנטרופיה

האנטרופיה היא החסם התחתון של מספר הסיביות לאות מקור שניתן להשיג בקידוד ללא עוותים. קיימות מספר נוסחאות לחישוב האנטרופיה אשר תלויות באופי המקור. אנטרופיה מסדר אפס עבור מקורות ללא זכרון מוגדרות על ידי הנוסחה:

$$H(u) = -\sum_{i=0}^{\alpha-1} P(u_i) \log P(u_i) \tag{B.1}$$

כאשר $P(u_i) \triangleq$ הסתברות הופעת הסימבול u_i , $u_i \in U$, $i = 0,1,\ldots,\alpha-1$, ועוצמת המקור U (מספר האותיות באלפבית שלו) היא α $(|U| = \alpha)$.

לקידוד בלוקים של סימבולים (וקטורים בעלי n סימבולים) הנוסחה המתאימה לאנטרופיה היא:

$$H(\underline{u}) = -\sum_{\underline{u}} P(\underline{u}) \log P(\underline{u}) \tag{B.2}$$

במקרה ש-$\underline{u} = (u_1,u_2)$ נרשום את הביטוי (A.2) בצורה:

$$H(\underline{u}) = -\sum_{i=0}^{\alpha-1} \sum_{j=0}^{\alpha-1} P(u_i,u_j) \log P(u_i,u_j) \tag{B.3}$$

עבור מקורות מרקוביים מסדר ראשון נוסחת האנטרופיה היא:

$$H(u_i/u_j) = -\sum_{i=0}^{\alpha-1} \sum_{j=0}^{\alpha-1} P(u_i,u_j) \log P(u_i/u_j) \tag{B.4}$$

כאשר

$$|U| = \alpha-1 \qquad i = 0,1,\ldots,\alpha-1, \quad u_i \in V$$

קיים משערך לאנטרופיה המבוסס על אלגוריתם זיו-למפל לקידוד אוניברסלי של סדרות נתונות, [9], אשר איננו דורש חישוב של ההסתברות להופעת הסימבולים השונים. הנוסחה היא:

$$\hat{H}(u) = \sum_{j=1}^{p+1} \lceil \log j \rceil / N \tag{B.5}$$

כאשר N: מספר הסימבולים שהמקור פלט (סופי).

$p+1$: מספר המחרוזות השונות שנוצרות לאחר הפיסוק של N הסימבולים,

וכאשר $\lceil x \rceil$ מציין את הערך השלם הגדול ביותר שכולל את x .

המונה של (5) מסמן את מספר הסיביות הדרוש להקצבת מספר מזהה לכל מחרוזת.

$\hat{H}(u)$ שואף לאנטרופיה האמיתית של המקור המתחשב במדויק בתכונות הסטטיסטיות של המקור כאשר N שואף לאינסוף; אין הוכחה פורמלית לטענה זו [9] .

$\hat{H}(u)$ מהווה חסם תחתון לתוצאת הדחיסה שניתן להשיג על ידי האלגוריתם זיו-למפל ללא תלות במימושו. נשתמש בחסם זה על מנת לקבוע את יכולת הדחיסה של האלגוריתם זיו למפל ביחס לאלגוריתמים אחרים לקידוד ללא עוות וכמו כן להשוואת תוצאות הדחיסה של המימושים השונים של האלגוריתם.

בטבלה מספר B.1 מוצגים ערכי האנטרופיה השונים שהתקבלו עבור 3 תמונות פנים עם יצוג של 8 סיביות לפיקסל.

טבלה B.1: ערכי האנטרופיה של תמונות המיוצגות על ידי 8 סיביות לפיקסל.

Table B.1: Entropy values of 8 bits/pixel pictures.

תמונה	$H(u)$	$H(\underline{u})^{(1)}$	$H(u_i/u_j)$	$\hat{H}(u)^{(2)}$	$\left(\frac{\text{סיביות}}{\text{פיקסל}}\right)$
1	7.0846	5.5862	4.2161	4.2735	
2	7.6439	5.9677	4.4448	4.5283	
3	6.5547	5.0594	3.7607	3.7652	
ממוצע	7.0944	5.5378	4.1405	4.1890	

(1) $\underline{u}$ הוא וקטור במימד 2.

(2) הערכים שהתקבלו עבור $\hat{H}(u)$ הם מקורבים בלבד מפני שגודל זכרון המחשב מגביל את החישוב של $\hat{H}(u)$ לרבע תמונה. התוצאה היא ממוצע של $\hat{H}(u)$ על 4 רבעי תמונה.

מתוך התוצאות שהוצגו בטבלה נובע:

1) קירבת ערכי האנטרופיה המותנית $H(u_i/u_j)$ לערכי האנטרופיה המשוערכת $\hat{H}(u)$ מעידה על ניצול התכונות הסטטיסטיות של התמונה על ידי האלגוריתם ZL.

2) במדידות הנ"ל מתקבל $\hat{H}(u) > H(u_i/u_j)$ מכיוון שהאלגוריתם לא התכנס ולכן הושג ערך $\hat{H}(u)$ שהוא גבוה מהאנטרופיה האמיתית. הסיבות לכך שהאלגוריתם לא התכנס הן:

א. הגבלה בזכרון שגרמה לכך שהאלגוריתם הופעל רק על רבעי תמונה שגודל כל אחת מהן 256×64 פיקסלים;

ב. התמונות אינן סטציונריות.

3) הדחיסה של האלגוריתם ZL טובה מהדחיסה על ידי קידוד Huffman עבור בלוקים של שני פיקסלים.

בטבלה מס' B.2 מופיעים ערכי האנטרופיה השונים שהתקבלו עבור אותן 3 תמונות פנים כאשר הפעם הן מיוצגות על ידי 4 סיביות לפיקסל.

טבלה B.2: ערכי אנטרופיה של תמונות המיוצגות על ידי 4 סיביות לפיקסל.

Table B.2: Entropy values of 4 bits/pixel pictures.

תמונה	$H(u)$	$H(\underline{u})$	$H(u_i/u_j)$	$\hat{H}(u)$	$(\frac{\text{סיביות}}{\text{פיקסל}})$
1	3.3958	2.3360	1.2986	1.3854	
2	3.9225	2.6283	1.3382	1.5192	
3	3.4264	2.2168	1.0560	1.2525	
ממוצע	3.5816	2.3937	1.2309	1.3857	

יחסי הדחיסה הממוצעים המתקבלים על פי הערך של $\hat{H}(u)$ הינם:

א. בתמונות המיוצגות על ידי 8 סיביות לפיקסל $8/4.1890 = 1.91$

ב. בתמונות המיוצגות על ידי 4 סיביות לפיקסל $4/1.3857 = 2.89$.

במקרה ב' יחס הדחיסה טוב יותר, הסיבה לכך היא שהאנטרופיה של 4 הסיביות הפחות משמעותיות גבוהה ($2.8 \frac{\text{סיביות}}{\text{פיקסל}}$) לעומת אנטרופיה של $\frac{\text{סיביות}}{\text{פיקסל}}$ 1.4 של 4 הסיביות היותר משמעותיות, זאת משום שהסיביות הפחות משמעותיות הן פחות קורלטיביות.

נספח ג': תאור פורמלי של מימוש מס' 1 של האלגוריתם ZL אשר הוצג בסעיף 2.1.1 בגוף התיזה

(0) התחל את עץ הפיסוק עם α ענפים כך שיכלול את כל הסימבולים של האלפבית.
קבע $j = \alpha$.

(1) החל מהמקום הנוכחי של המחוון בסדרת הכניסה, מצא את המחרוזת הארוכה ביותר אשר שייכת לעץ, כלומר כאשר חיפוש המחרוזת בעץ הפיסוק מגיע לעלה.

(2) פלוט את הקוד של המספר המזהה של המחרוזת שנמצאה, דהיינו את המספר המזהה של העלה.

(3) הוסף α עלים חדשים לעץ בצומת האחרון שאליו מגיעים בתהליך הפיסוק(*).
הקצב בסדר הנתון לכל עלה חדש את הערך j (*). בין הקצבה להקצבה קבע $j = j+1$.
(*) פרט לעלה שאליו ניתן להגיע דרך אותו סימבול כניסה של הצומת שלפניו.
לעלה זה הקצב את אותו מספר מזהה של הצומת שממנו נוצר העלה.

(4) קדם את המחוון של סדרת הכניסה מעבר למחרוזת החדשה. חזור לשלב (1) אם המחוון לא מצביע על סוף הבלוק.

(5) בסוף הבלוק חזור לשלב (0).

(*) שים לב שהוספת α עלים חדשים לצומת אקויוולנטית להוספת $\alpha-1$ עלים חדשים לעץ כולו.

נספח ד': מימוש המפענח

באופן עקרוני ביצוע הפיענוח נעשה בעזרת עץ פיסוק בינרי (עבור $\alpha = 2$) שנבנה בו זמנית עם עץ הפיסוק שבמקדד. פענוח בדרך זו אינו יעיל כי זמן החיפוש ארוך. על מנת שזמן החיפוש יהיה $O(1)$ נשתמש בטבלה, שאף היא נבנית בו זמנית עם עץ הפיסוק שבמקדד. מפתח החיפוש בה היא מילת קוד. בכניסה למפענח מתקבלות מילות קוד באורך משתנה ללא סימון מיוחד המבדיל ביניהן.

מבנה הטבלה

CW	L	SEQUENCE
	1	0
	1	1

CW: מסמן מילת קוד. אין צורך לאחסן את מילות הקוד כי מיקומן הסידורי בטבלה מתאים לערכן המספרי.

L : אורך המחרוזת (מספר סיביות) השייכת למילת הקוד.

SEQUENCE: המחרוזת המתאימה למילת הקוד.

שלבי הפיענוח

בתאור שלבי הפיענוח נעזר בסימונים הבאים:

RCW : מילת הקוד שהתקבלה.

NCW : מילת הקוד החדשה שנוצרה בעידכון הטבלה.

FPT : המקום הפנוי הראשון בטבלה.

LB : הסיבית האחרונה במחרוזת מסויימת.

(0) התחל את הטבלה עם הסיביות המתאימות למילות הקוד 0 ו-1 (ראה ציור). קבע $FPT = 2$.

(1) קרא מילת קוד (RCW) (אורך מילת הקוד RCW מתקבל מחישוב מראש על ידי: $\lceil \log(FPT) \rceil$).

(2) פלוט את המחרוזת (סדרת סיביות) המתאימה למילת הקוד שהתקבלה.

(3) שרשר סיבית נוספת (משמאל) בעלת ערך זהה לזה של LB, במקום בטבלה המתאים ל-RCW (מילת הקוד שהתקבלה). קדם את הערך של L ב-1 ($L \leftarrow L+1$).

(4) הוסף מילת קוד חדשה (NCW) לטבלה על ידי:

(A) העתקה של סדרת סיביות השייכת ל-RCW למקום FPT;

(B) עידכון ערך L במקום FPT (ערך L של NCW שווה לזה של RCW לאחר העידכון);

(C) היפוך ערך LB ($LB \leftarrow \overline{LB}$) בסדרת הסיביות השייכת ל-NCW;

(D) קידום FPT ב-1 ($FPT \leftarrow FPT+1$).

(5) חזור לשלב (1).

<u>הצעה ליעול המפענח</u>

על מנת לחסוך בכמות הזכרון הדרושה לפיענוח הוצע שימוש בשתי טבלאות. הראשונה תהיה זהה לזו שתוארה במימוש המקורי, אך צרה ממנה. רוחב הטבלה יקבע כך שרוב המחרוזות באורך בינוני (ביחס למחרוזות האחרות) תכללנה בטבלה זו. הטבלה השניה תכלול רק את המשכן של אותן מחרוזות ארוכות. לכן, הגישה לטבלה זו תעשה רק בתנאי שהמחרוזות של מילות הקוד שהתקבלו תהיינה ארוכות **יותר** מרוחב הטבלה הראשונה.

בדרך זו נמנעים מלקצר בצורה משמעותית את האורך המכסימלי על המחרוזת ומנצלים טוב יותר את כמות הזכרון.

נספח ה': תיאור אלגוריתם הקידוד עם הגבלת אורך המחרוזת

שלבי הפיענוח מתבצעים בעזרת טבלה המאפשרת פיענוח מהיר אך המוגבלת באורך המכסימלי של המחרוזות שניתן לאחסן בה. על מנת לאפשר קידוד ופענוח עם אורך מחרוזות מוגבל, דרושים שינויים קלים באלגוריתם הקידוד והפענוח.

בתאור שלבי הקידוד נעזר בגודל $\stackrel{\Delta}{=} \ell_c$ אורך המחרוזת שנוצרה בתהליך הפיסוק.

ℓ_{max} הינו האורך המקסימלי שהוקצב למחרוזות.

<u>שלבי הקידוד</u> (השינויים ביחס לאלגוריתם המקורי מסומנים בקו מתחת לשורה).

(0) התחל את העץ הפיסוק עם 2 ענפים כך שיכלול את הסימבולים '0' ו-'1'.

קבע $j = 2$. <u>קבע $\ell_c = 0$</u> ;

(1) החל מהמקום הנוכחי של המחוון בסדרת הכניסה, מצא את המחרוזת הארוכה ביותר אשר שייכת לעץ הפיסוק. <u>קבע ℓ_c כאורך המחרוזת החדשה;</u>

(2) פלוט את הקוד של המספר המזהה של המחרוזת שנמצאה, דהיינו את המספר המזהה של העלה;

(3) <u>אם $\ell_c < \ell_{max}$</u> אזי: הוסף 2 עלים חדשים לעץ בצומת האחרון שאליו מגיעים בתהליך הפיסוק. הקצב לעלה שאליו ניתן להגיע דרך אותו סימבול כניסה של הצומת שלפניו את אותו הערך של הצומת. לעלה השני קבע ערך j . קבע $j = j + 1$.

(4) <u>קבע $\ell_c = 0$</u> ;

(5) קדם את המחוון של סדרת הכניסה מעבר למחרוזת החדשה. חזור לשלב (1) אם המחוון לא מצביע על סוף הבלוק;

(6) בסוף הבלוק חזור לשלב (0).

נספח ו': השוואה של הקצבת מילות הקוד בשני מימושים של האלגוריתם ZL

נסמן ב-L^T את מספר הסיביות הדרוש לקידוד של $p+1$ מחרוזות שמתקבלות במימוש המבוסס על עץ פיסוק מדרגה α. נסמן ב-L^B את מספר הסיביות הדרוש במימוש המבוסס על עץ פיסוק בינרי תוך שמירת גבולות ה-byte, כדי לקודד את אותן $p+1$ המחרוזות.

<u>טענה</u> $L^B < L$ עבור $\alpha \geq 3$

כאשר:

$$L^T = \sum_{j=1}^{p+1} \lceil \log((\alpha-1)j) \rceil \tag{F.1}$$

$$L^B = \sum_{j=1}^{p+1} \{ \lceil \log(j + \sum_{i=1}^{j-1} EC_i) \rceil + EC_j \} \tag{F.2}$$

p : מס' המחרוזות שהתקבלו לאחר הפיסוק.

<u>הוכחה</u>

נניח ש-$k/2$ = $\langle EC_j \rangle$, $k = \lceil \log_2 \alpha \rceil$.

לאחר הצבה $k/2$ ב-(F.2) נקבל:

$$L^B = \sum_{j=1}^{p+1} \lceil \log(j + \sum_{i=1}^{j-1} k/2) \rceil + k/2 \} \tag{F.3}$$

כמו כן נשתמש באי-השוויון

$$\lceil \log j \rceil \leq \log 2j \tag{F.4}$$

אזי נרשום עבור (F.1)

$$L^T \leq \sum_{j=1}^{p+1} \log(2(\alpha-1)j) = \sum_{j=1}^{p+1} \log j + (p+1)(\log 2 + \log(\alpha-1)) \tag{F.5}$$

הצבה של (F.4) ב-(F.3) נותנת:

$$L^B \leq \sum_{j=1}^{p+1} \log(2j + (j-1)k) + (p+1)\frac{k}{2} \tag{F.6}$$

נרשום את הארגומנט של ה- log בצורה הבאה:

$$j(2+k) - k$$

נגדיל שוב את אי-השוויון ונרשום את (F.6) בצורה הבאה:

$$L^B < \sum_{j=1}^{p+1} \log j(2+k) + (p+1)\frac{k}{2} \qquad (F.7)$$

$$= \sum_{j=1}^{p+1} \log j + (p+1)\left(\log(2+k) + \frac{k}{2}\right)$$

על מנת לבדוק את הביטויים נשווה את הביטויים (F.5) ו-(F.7) : $L^B \gtrless L^T$

$$\sum_{j=1}^{p+1} \log j + (p+1)\left(\log(k+2) + \frac{k}{2}\right) \gtrless \sum_{j=1}^{p+1} \log j + (p+1)(1+\log(\alpha-1)) \qquad (F.8)$$

לאחר הצימצום נותר לבדוק:

$$k/2 + \log(2+k) \lesseqgtr 1 + \log(\alpha-1) \qquad (F.9)$$

נציב $k = \lceil \log \alpha \rceil$ ב-(F.9):

$$\lceil \log \alpha \rceil/2 + \log(2+\lceil \log \alpha \rceil) \gtrless 1 + \log(\alpha-1) \qquad (F.10)$$

לאחר צימצום נוסף נקבל:

$$\log(2 + \lceil \log \alpha \rceil) < 1 + \log(\alpha-1)/2 \qquad (F.11)$$

עבור $\alpha \geq 3$.

מ.ש.ל.

נספח ז': הסתברות הופעת מצבי שרשרת מרקוב

שרשרת מרקוב המתארת את המקור Y של מודל יצירת תמונה מופיעה בסעיף 4.1.4.

נסמן ב-$\pi(j)$ את הסתברות הופעת המצב j במצב המתמיד.

נוכיח את הטענה:

$$\pi(j) = 1/16 \quad \text{לכל} \quad j \;, \quad j \in e = \{0,1,\ldots,15\}$$

הוכחה

בהוכחה נעזר במספר משפטים בסיסיים על שרשראות מרקוב המופיעים בספרו של E. Cinlar [23].

טענה: כל המצבים בשרשרת מרקוב Y הם נשנים (recurrent).

הוכחת הטענה: שרשרת Y לא ניתנת לצימצום מפני שניתן להגיע לכל המצבים מכל המצבים.

נסמן ב-T את הפעם הראשונה בה השרשרת מבקרת במצב j כלשהוא.

מתוך הסתכלות בדיאגרמת המצבים ניתן לקבוע שעבור כל j מתקיים:

$$j \in e, \quad P\{\, T < \infty \,/\, y_0 = j \,\} = 1$$

מכאן שהמצב j על פי ההגדרה הוא מסוג נשנה (recurrent). מ.ש.ל.

על מנת לחשב את ההסתברויות הגבוליות $(n \to \infty)$ נעזר במשפט הבא:

משפט: עבור שרשרת מרקוב שאינה ניתנת לצימצום ואינה מחזורית, מתקיים שכל המצבים הם נשנים (recurrent) אסם למערכת של המשוואות הליניאריות הבאות:

$$\pi(j) = \sum_{i \in e} \pi(i)\, P(i,j) \qquad j \in e$$

$$\sum_{j \in e} \pi(j) = 1$$

יש פתרון π, שהוא יחיד ומקיים:

$$\pi(j) = \lim_{n \to \infty} P^n(i,j)$$

לכל $i\,,\, j \in e$.

כאשר שרשרת מרקוב בעלת מספר סופי של מצבים נקבל את $\pi(j)$ על ידי:

$$P\{y_1 = j\} = \pi(j)$$

אזי:

$$P\{y_n = j\} = \sum_{i \in e} \pi(i)\, P^n(i,j) = \pi(j)$$

לכל $n \in N$.

מכיוון שמצבי השרשרת הם נשנים, ועל סמך המשפט הנ"ל מספיק לחשב את $P\{y_1=j\}$ כאשר y_i הינו המצב של השרשרת ברגע i. חישוב מערכת המשוואות הלינאריות נותן $\pi(j) = 1/16$.

ומכאן:

$$\forall\ i, \quad j \in e \quad , \quad \pi(j) = \lim_{n\to\infty} P^n(i,j) = 1/16$$

מ.ש.ל.

נספח ח': חישוב אנטרופית המקור Y (מודל ליצירת תמונה)

מודל יצירת תמונה מבוסס על מקור מרקובי מסדר ראשון Y. אנטרופית המקור נתונה

על ידי: $H(y_n/y_{n-1})$.

נוכיח את הטענה: $H(y_n/y_{n-1}) = H(x)$

כאשר $H(x)$ היא האנטרופיה מסדר אפס של המקור X שבמודל.

<u>הוכחה</u>

$$y_n = (y_{n-1}+x_n) \bmod 256$$

$$H(y_n/y_{n-1}) = -\sum_{i=0}^{2^N-1} \sum_{j=0}^{2^N-1} P(y_n=j,\ y_{n-1}=i)\cdot \log P(y_n=j/y_{n-1}=i)$$

$$= \sum_{i\epsilon e} \sum_{j\epsilon e} P(y_n=j/y_{n-1}=i)\ P(y_{n-1}=i)\ \log P(y_n=j/y_{n-1}=i)$$

כאשר 2^N: מספר המצבים בשרשרת.

e: אוסף המצבים בשרשרת.

על ידי החלפת סדר הסכימה נקבל:

$$H(y_n/y_{n-1}) = -\sum_{i\epsilon e} P(y_{n-1}=i) \sum_{j\epsilon e} P(y_n=j/y_{n-1}=i)\ \log P(y_n=j/y_{n-1}=i)$$

עבור $i \epsilon e$ קבוע, ולכל j, $j \epsilon e$, $P(y_n=j/y_{n-1}=i)$ נותן את הערכים של הסתברות המעבר מהמצב i לכל מצב j (השורה ה-i-ית במטריצת המעבר).

לכן, עבור i קבוע, $i \epsilon e$, הביטוי

$$-\sum_{j\epsilon e} P(y_n=j/y_{n-1}=i)\ \log P(y_n=j/y_{n-1}=i)$$

שווה לביטוי:

$$-\sum_{k\epsilon X} P(x_n=k)\ \log P(x_n=k) \overset{\Delta}{=} H(x)$$

כי:

$$P(y_n=j/y_{n-1}i) = P(x_n=k)$$

כאשר:

$$k \epsilon A = \{0,\pm\Delta,\pm2\Delta,\pm3\Delta\}$$

אזי:

$$\sum_{i\epsilon e} P(y_{n-1}=i)\ H(x) = H(x) \sum_{i\epsilon e} P(y_{n-1}=i) = H(x)$$

מ.ש.ל.

נספח ט': מודל ליצירת תמונה - הסתברויות לפליטת $B_\ell(i)$

בטבלה הבאה מוצגות ההסתברויות לפליטת '1', $P_i^\ell(1)$ של $B_\ell(i)$, לכל מצב i,
עבור הייצוג הבינרי של ערכי המצבים. ℓ מסמן מישור סיבית.
ההסתברות לפליטת '0', $P_i^\ell(0)$ נתונה על ידי $P_i^\ell(0) = 1 - P_i^\ell(1)$.

מצב i	$P_i^0(1)$ (ℓ=0)	$P_i^1(1)$ (ℓ=1)	$P_i^2(1)$ (ℓ=2)	$P_i^3(1)$ (ℓ=3)
0	$P_1+P_2+P_3$	$P_1+P_2+P_3$	$P_1+2P_2+P_3$	$2P_1+2P_3$
1	P_2+P_3	P_2+2P_3	$P_1+2P_2+P_3$	$1-(2P_1+2P_3)$
2	P_3	P_2+2P_3	$1-(P_1+2P_2+P_3)$	------------ (*)
3	0	$P_1+P_2+P_3$	$1-(P_1+2P_2+P_3)$	
4	0	$1-(P_1+P_2+P_3)$	------------ (*)	
5	P_3	$1-(P_2+2P_3)$		
6	P_2+P_3	$1-(P_2+2P_3)$		
7	$P_1+P_2+P_3$	$1-(P_1+P_2+P_3)$		
8	$1-(P_1+P_2+P_3)$	------------ (*)		
9	$1-(P_2+P_3)$			
10	$1-P_3$			
11	1			
12	1			
13	$1-P_3$			
14	$1-(P_2+P_3)$			
15	$1-(P_1+P_2+P_3)$			

* ערכי ההסתברויות של המצבים הבאים הינם מחזוריים.

נספח י : תצלומים של קבוצת תמונות הבדיקה

I) תמונות מקוריות

א. יצוג על ידי 4 סיביות לפיקסל

תמונה 1
Picture 1

תמונה 2
Picture 2

תמונה 3
Picture 3

ב. תמונות מקוריות לאחר תוספת Ordered Dither.

תמונה 1
Picture 1

תמונה 2
Picture 2

II) תמונות לאחר עבוד קדם או הפעלת שיטות קידוד שונות. כל התמונות מיוצגות על ידי 4 סיביות בתוספת Order Dither.

א. דצימציה מרחבית 2:1 - Spatial Decimation

תמונה 1
Picture 1

תמונה 2
Picture 2

ב. דצימציה מרחבית 4:1 Spatial Decimation

תמונה 1
Picture 1

תמונה 2
Picture 2

ג. קידוד דלטה מסתגל (ADM).

תמונה 1
Picture 1

תמונה 2
Picture 2

ד. התמרת הדמרד - Hadamard Transform.

פרטי ביצוע ההתמרה

גודל בלוק ההתמרה: 4 × 4 פיקסלים.

מס' מקדמים מאופסים: 7

כימוי אחיד לכל המקדמים, א) למקדם הראשון, (C_1) הוקצבו 6 סיביות

ב) לכל אחד מהמקדמים הוקצבו 4 סיביות.

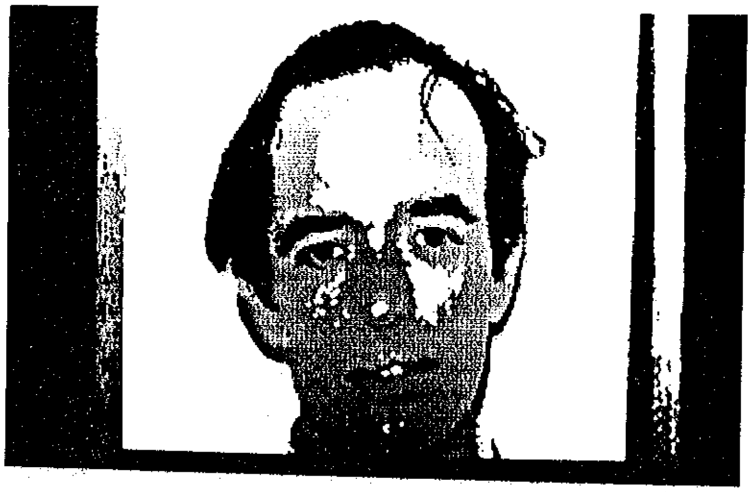
תמונה 1
Picture 1

תמונה 2
Picture 2

III) שיחזור הדרגתי של התמונה - Progressive reconstruction of the picture.

תמונה 2
Picture 2

סיבית אחת (MSB)
One bit



שתי סיביות
Two bits

שלוש סיביות
Three bits

ארבע סיביות
Four bits.

REFERENCES

רשימת מקורות

1. J. Ziv and A. Lempel, "Compression of Individual Sequences Via Variable Rate Coding." IEEE Trans. Inform. Theory. Vol. 24, pp. 530-536, Sep. 1978.

2. A. Habibi, "Survey of Adaptive Image Coding Techniques", IEEE Trans. Commun., Vol. COM-25, pp. 1275-1284, Nov. 1977.

3. A.N. Netravali and J.D. Limb, "Picture Coding: A Review", Proc. IEEE, Vol. 68, pp. 336-406, Mar. 1980.

4. A.K. Jain, "Image Data Compression: A Review", Proc. IEEE, Vol. 69, pp. 349-389, March, 1981.

5. R.C. Gonzales and P. Wintz, Digital Image Processing. Reading, Mass.: Addison-Wesley, 1977.

6. W.K. Pratt, Digital Image Processing. New York: Wiley, 1978.

7. A. Rosenfeld and A.C. Kak, Digital Picture Processing - 2nd. ed., New York: Academic Press, 1982.

8. R.J. McElice, The Theory of Information and Coding. Reading, Mass.: Addison-Wesley, 1977.

9. J. Ziv, Private Communication.

10. C.N. Judice, J.F. Jarris and W.H. Ninke, "Using Ordered Dither to Display Continuous Tone Pictures on an AC Plasma Panel", Proc. S.I.D., Vol. 15/4, pp. 161-169, Fourth Quarter, 1974.

11. S. Even, "On Information Lossless Automatic Of Finite Order", IEEE Trans. Computers, Vol. EC-1, pp. 561-569, Aug. 1965.

12. D. Knuth, The Art of Computer Programming, Vol. 3 Sorting and Searching. Reading, Mass.: Addison Wesley, 1968.

13. D. Malah, Private Communication.

14. R.G. Gallager, Information Theory and Reliable Communication. New York: Wiley, 1968.

15. H.J. Narasimma and A.H. Peterson, "On the Computation of the Discrete Cosine Transform, IEEE Trans. Commun., Vol. COM-26, pp. 934-936, June 1978.

References (Cont.) רשימת מקורות (המשך)

16. M. Gardner, "Mathematical Games," Scientific American, pp. 124-133, Dec. 1976.

17. N. Scheinberg and D.L. Schilling, "Techniques for Correcting Transmission Errors in Video Adaptive Delts Modulation Channels " IEEE Trans. Commun., pp. 1064-1069, Sep. 1976.

18. W. Chen and W.K. Pratt, "Scene Adaptive Codes", IEEE Trans. Commun., Vol. COM-32, pp. 225-232, March 1984.

19. W.K. Pratt, J. Kane and H.C. Andrews, "Hadamard Transform Image Coding", Proc. IEEE, Vol. 57, pp. 58-68, Jan. 1969.

20. H.J. Landau and D. Slepian , "Some Computer Experiments in Picture Processing for Bandwith Reduction", Bell Syst. Tech. J., Vol. 50, pp. 1525-1540, May-June 1971.

21. H. Murakami, Y. Hatori and H. Yamamoto, "Comparison Between DPCM and Hadamard Transform Coding in the Composite Coding of the NTSC Color TV Signal", IEEE Trans. Commun., Vol. COM-30, pp. 469-479, March 1982.

22. N.S. Jayant and P. Noll, Digital Coding of Wareforms. Englewood Cliff, New Jersey: Prentice Hall, 1984.

23. E. Cinlar, Introduction to Stochastic Processes. Englewood Cliff, New Jersey: Prentice Hall, 1975.

EFFICIENT DATA-BASE STORAGE OF IMAGES BY COMPRESSION TECHNIQUES

Research Thesis
Submitted in Partial Fulfillment of the Requirements
For the Degree of
Master of Science
in Electrical Engineering

By

Ariel Lejtman

Submitted to The Senate of The Technion - Israel Institute of Technology

Heshvan 5745 HAIFA November 1984

This research paper was carried out in the faculty of Electrical Engineering under the supervision of Professor David Malah.

My sincere expression of gratitude to Professor D. Malah, for his devoted guidance, vast contribution and full involvement in all stages of this work.

I wish to acknowledge thankfully my indebtedness to Professor Jacob Ziv for his important contribution in the guidance of this research.

Many thanks to Zippi Portnoy and to Yoram Or-Hen for their generous help at the Signal Processing Laboratory.

To Ilit who encourages me, helped me, smoothed the rough edges: thank you.

To my parents

and to Ilit

CONTENTS

Contents (Cont.)

ABSTRACT

In this work new implementations - for picture coding - of the Ziv-Lempel algorithm are presented and analyzed.

The Ziv-Lempel Universal Coding algorithm is an information-lossless compression scheme which does not require prior knowledge of the statistical properties of the source; at the same time, the rate achieved by this algorithm approaches the entropy of a given stationary source.

This algorithm has been applied to different sources; however, there have been up to now no reports of its usage in picture coding.

The goal of the present research work is to develop new implementations of the algorithm with the view of achieving efficient picture coding, i.e., maximal compression with minimal complexity.

In the three new versions of the algorithm implementation presented in this work the coding is done with the help of a parsing tree, and the decoding by the means of a look-up table. This guarantees a fixed and equal decoding time for all code words, and thus a rapid reconstruction of the pictures.

The first version separates the picture data into multiple bit planes (according to the number of bits/pixel used to represent the picture) and the coding process in done on each plane separatly. In order to obtain good compression results the coding is carried out on pictures represented by their GRAY code equivalent value. In this representation of the picture a high correlation between sucessive bits of the most significant bit planes is achieved. In fact, this scheme shows an improved bit/pixel compression results in comparison with existing implementations for this algorithm. Furthermore, it becomes possible to reconstruct pictures gradually and thereby to speed up the searching process in data-bases.

The second version works with a fixed coding tree at both the encoder and the decoder. This coding tree, called universal tree, is constructed in advance using a set of characteristic pictures. By this scheme a fixed length codeword and enhanced invulnerability to noisy channels is obtained. When implementing this version by parsing the picture according to its original alphabet, it was found that the compression achieved varies with the statistical properties of the pictures. Since pictures are not stationary the performance of the algorithm is degradated.

In the third version, the two previous ones are combined, i.e., the coding is done with the help of universal trees on each bit plane separetely. In addition to the advantages already achieved by the two other versions, this coding scheme was found to be more robust to variation in picture characteristics.

A picture production model, based on a first order Markov source is introduced, enabling us to explain and to evaluate the performance of the proposed implementations. For the first version a good correspondence was found between the model and real pictures. However a more complex model is needed to explain the reasons of the different sensibilities of the second and third versions of the algorithm.

By coding 8 bit/pixel portrait pictures by the Ziv-Lempel algorithm a compression ratio of no more than 2:1 can be achieved. In order to improve this ratio the coding is done on 4 bit/pixel pictures. A compression ratio of up to 3:1 can be obtained and the quality of these pictures is good indeed. It can further improved by adding ordered dither.

Still further improvement of the compression ratio can be achieved by combining this algorithm with other ones. Among the coding techniques we examined, it was found that Hadamard Transform Coding offers the most satisfactory results. It was applied to 8 bit/pixel pictures and after zeroing the low variance coefficients and quantizing the remaining ones, the Ziv-Lempel algorithm was used. By this method a compression of up to one bit per pixel is obtained, i.e. a compression ratio of 8:1; the quality of the reconstructed picture being very good.