**עיבוד וניתוח וידיאו – פרויקט גמר**

מטרת הפרויקט היא יצירת סרטון של אדם על רקע נתון ועקיבה אחריו כשהקלט הוא סרטון רועד של אדם ההולך על רקע סטטי וביצוע השלבים הבאים:

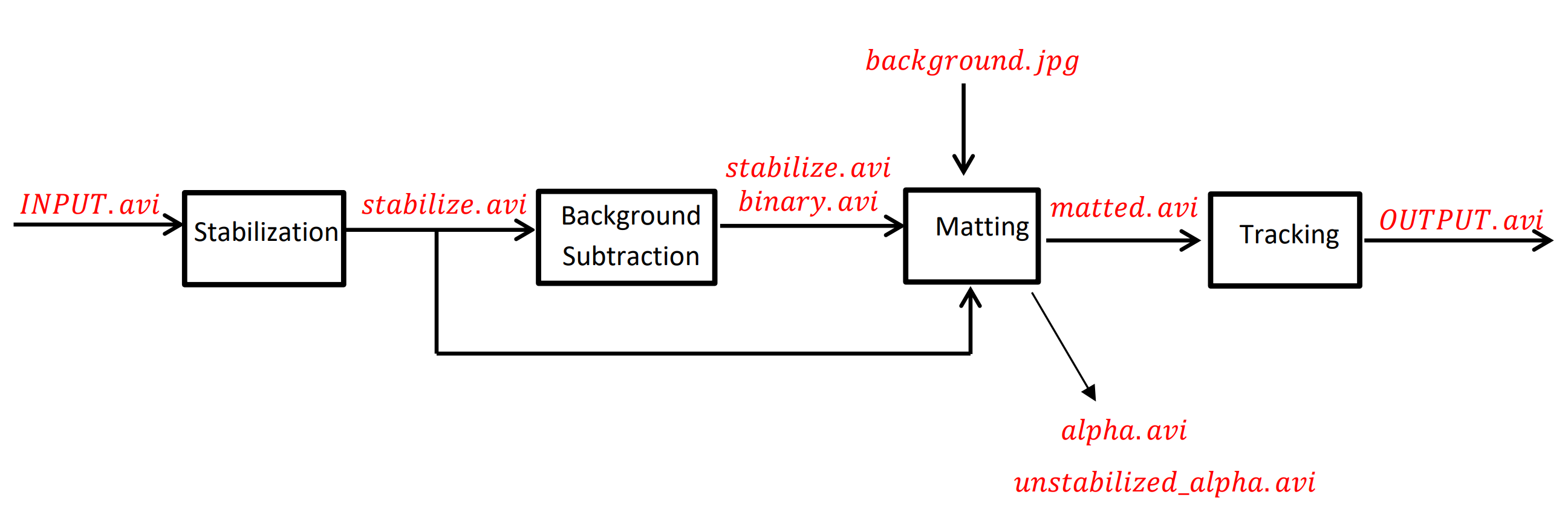
- ייצוב הסרטון (Stabilization)

- הסרת הרקע ובידוד האדם מהרקע (Background Substruction)

- יצירת סרטון Matted סרטון בוא האדם הולך על רקע חדש. זאת ע"י יצירת סרטון אלפא המגדיר את "היחס" בין האדם לבין הרקע החדש.

- עקיבה אחר האדם בסרטון ה-Matted.

סכמת הפרויקט כפי שצורפה בתדריך:



**קובץ runme ואופן השימוש:**

כדי להריץ את התוכנית על המשתמש להריץ את הקובץ “runme.py”.

הפונקציה runme מריצה את ארבע הפונקציות השונות stabilize, bgs, matting, track.

אם המשתמש רוצה להריץ פונקציה ספציפית, עליו לוודא שהקלט שלה קיים ולהוציא מהערה את השורה המתאימה בחלק “run blocks individually” בקובץ הrunme. בפונקציה tracking יכול המשתמש לשנות את המשתנה user\_input להיותTrue ואז יוכל לבחור את גודל ריבוע העקיבה.

עבור ריצה מלאה התוכנית מדפיסה כל התחלה וסיום של בלוק ובמהלך ריצת הבלוק מדפיסה את אחוז העבודה שכבר התבצע.

בריצה חלקית יודפס אחוז העבודה בלבד.

\* חשוב לשים לב שזמן הריצה בהקלטה גבוה משמעותית מזמן הריצה ללא הקלטה. זמן הריצה המקורי הוא 630 שניות (עם הקלטה 1326).

נפרט על הבלוקים השונים בפרויקט, המטרה של כל בלוק, הטכניקות והפונקציות בהן השתמשנו, השיקולים בבחירת הדרך והקוניפיגורציות והתוצר הסופי:

**ייצוב – Stabilization**

מטרת חלק זה היא לייצב את סרטון הקלט הרועד. מהסתכלות על הסרטון המקורי ניתן לראות שתזוזת המצלמה מעגלית במובן מסוים. בהתחלה חשבנו לעבוד עם סקריפט הייצוב שכתבנו במהלך הקורס, שניסינו למדל את הרעד ע"י טרנספורמציה מסוימת מאלו שנלמדו הבנו שהתנועה מורכבת יותר והחלטנו לנסות להשתמש בפונקציה המובנית לחישוב ה optical flow של Opencv (cv2.calcOpticalFlowPyrLK). הפונקציה דורשת מאיתנו בין השאר כקלט, לזהות אזורים בתמונה שיסיעו לעקיבה, כדי שתתקבל תוצאה טובה יותר עבור ה optical flow. לצורך כך השתמשנו בפונקציה המובנית cv2.goodFeautresToTrack.

לאחר מכן, אנו רוצים ליצור טרנספורמציה אפינית בין תמונת המקור לתמונה בפריים העוקב בהתאם ל OF, (באופן דומה למה שלמדנו בקורס ומימשנו בעצמנו בתרגיל בית 2). לצורך כך השתמשנו בפונקציה estimateAffinePartial2D. המטריצה שקיבלנו (נסמן ב -m), ביצענו מספר מניפולציות כדי לשפר את התוצאות, ומתוכה הוצאנו את מידע משוערך לגבי הסיבוב של הפריים בכל ציר (אנכי, אופקי ועומקי).

לאחר מכן יצרנו וקטור יחיד שמייצג את שלושת התזוזות לעיל (בשם trajectory). כדי לשפר את התוצאות ביצענו החלקה על וקטור ה trajectory בעזרת שיטת moving averages\*, ושוב חילצנו את התזוזות בכל ציר.

בנינו שוב את המטריצה m (בעצם בתהליך הפוך לאחר ביצוע ההחלקה) וביצענו warp בין הפריים והמטריצה שמייצגת את התזוזה על פי ה OF. לאחר מכן ביצענו מניפולציות פשוטות כדי לוודא שגודל הפריים נשאר קבוע ביחס לקלט.

- להרצת חלק זה על המשתמש לקרוא לפונקציה “stabilize( input\_video)”.

**הסרת רקע – Background Subtraction**

מטרת חלק זה היא ליצור סרטון בינארי אשר החלק הלבן בו מייצג את הדמות, והשחור את הרקע. כמו כן אנו יוצרים סרטון extracted בו הרקע שחור והדמות בצבע המקורי.

לצורך מימוש ה-Background Subtraction ניסינו 3 שיטות שונות. החלקת חציון, KDE ו- KNNSubtractor של opencv.

החלקת חציון:

ביצענו חישוב של הרקע בעזרת חציון על הסרטון, בכך ע"י ריצה ראשונית על הסרטון יצרנו תמונת רקע. חיסרנו את הרקע המחושב מכל פריים ובחרנו ערך סף שיעיד על הבדל גדול בין הרקע לבין ערך הפיקסל כלומר שמדובר בדמות. ככל ששינינו את הערך הזה נכנסו רעשים שונים ביניהם של הדלתות ברקע הסרטון למשל. אנחנו חושבים שהסיבה לכך היא שלא נכון להניח כאן Constant Brightness כלומר ייתכן שבגלל תזוזת המצלמה התאורה משתנה ויהיה שוני בין ערכי הרקע בפריימים השונים. ניסינו דרכים רבות לשפר את התוצאות של השיטה הנ"ל אך ראינו לבסוף שהתוצאות שלה גרועות משמעותית משתי השיטות האחרונות שניסינו ולכן לא המשכנו לממש אותה.

KDE:

השיטה השנייה שבה ניסינו לבצע הסרת רקע היא על ידי שימוש ב KDE, לצורך כל בחרנו סקריבלים על גבי התמונה שמייצגים את הדמות, ואת הרקע, ויצרנו פונקציית הסתברות בהתאם. לאחר מכן בכל פריים, עבור כל פיקסל, בדקנו האם הסבירות שלו להשתייך לרקע גדולה מהסבירות להשתייך לדמות, אם כן השחרנו אותו, ואחרת "הלבנו" אותו. על אף שביצענו מספר מניפולציות על גבי השיטה הנ"ל ( connected components, פעולות מורפולוגיות, הוספת סף וכו') לא הצלחנו להגיע לתוצאות טובות יותר, בפרט תהליך חישוב ה KDE הוא ארוך מאוד, ולכן הגענו לתוצאות בינוניות בלבד עם זמן ריצה ארוך ולכן בחרנו שלא להמשיך עם השיטה הנ"ל.

KNNSubtractor:

השיטה השלישית שבה היא עבדנו היא על ידי מודול ה background subtraction מבוסס K- nearest neighbors של OpenCV. המימוש בשיטה זאת דומה להחלקת חציון, אולם מבצע אינטרפולציה מבוססת KNN. את המימוש שלנו נחלק ל3 חלקים:

1. איסוף מידע – כדי לדאוג של-subtractor יש מספיק מידע על הסרטון, כדי שיוכל לבצע הסרת רקע בצורה טובה, אנו מתחילים משלב preprocess שבו אנחנו מריצים את הסרטון מספר פעמים, מבלי לבצע חיסור רקע, כדי לתת כמה שיותר מידע ל subtractor. לאחר ניסוי של מספר שונה של איטרציות, החלטנו לבצע 2 איטרציות של מעבר מוקדם (בהתאם לשיקולים של תוצאות וזמן ריצה).

2. Masking – גם לאחר שקיבלנו את המסכה של ה KNN ראינו שהרקע לא הוסר לחלוטין. מכיוון שמספר חלקים ברקע, קרובים בצבעם לדמות, וכן זזים (בגלל שהסרטון אינו מיוצב באופן מושלם). לצורך כך, עברנו לערוצי צבע HSV, והסתכלנו על הערוץ V. תחת הנחת הפריור שצבעי הדמות שזזה שונים מצבעים של אובייקטים ברקע, הגבלנו את תחומי הפיקסלים ב HSV (חשוב לציין שזו הגבלה כללית, לא בחרנו ערכי פיקסלים ספציפיים), והסרנו את חלקי תמונה שאינם בטווח הנבחר. לצורך שיפור התוצאות ביצענו את המיסוך הנ"ל פעמיים.

3. Morphological Transformation and Connected Components.

גם לאחר המיסוך, עדיין קיבלנו חלקים בתמונה שלא שייכים לדמות. אולם ראינו, שהדמות הינה החלק המשמעותי יותר במסכה שנוצרה. לכן כדי להסיר חלקים לבנים שלא מחוברים לדמות השתמשנו בפעולות מורפולוגיות כמו open ו close, על מנת להסיר רעשים, וכן לאחד חלקים מתוך הדמות לכדי רכיב אחד. לאחר מכן השתמשנו בפונקציה המובנית cv2.findContours, שמזהה רכיבים בתמונה ומסמנת אותם. ובכל פריים בחרנו לקחת את הרכיב הגדול ביותר (תחת ההנחה שבשלב זה לאחר שביצענו מיסוך קודם, הרכיב המהותי ביותר הוא אכן הדמות).

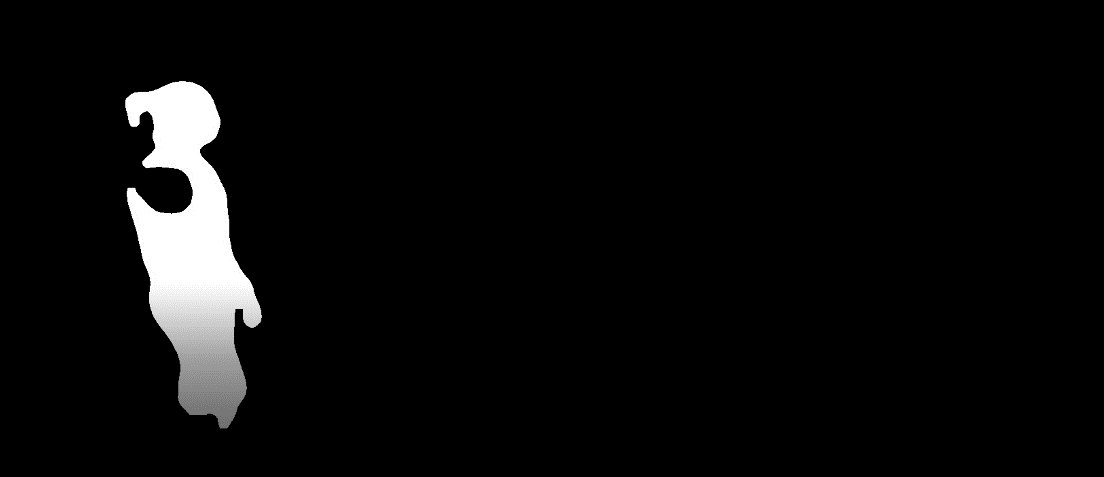
לאחר שלושת השלבים שתיארנו לעיל, קיבלנו את המסכה הבינארית הסופית שאיתה נעבוד. ובעזרתה חישבנו גם את הסרטון extracted.

לאורך הסרטון גם ביצענו מדי פעם החלקה בעזרת medianBlur, על מנת לקבל גבולות חלקים יותר וכן לצמצם רעשים.

בתוצאה בחלק זה ישנן בעיות מסוימות, בהן היעלמות חלקית של הנעליים בפריימים מסוימים והכנסת רעשי רקע (דלתות ופוסטרים) כ-trade-off להיעלמות הדמות. ניסינו דרכים שונות למצוא את עמק השווה האידיאלי, ביניהם ע"י החלקת התמונה ע"י מסננים שונים כשבסוף נשארנו עם הערכים הנוכחיים.

דוגמאות לtrade-off הזה:

שימוש בMOG subtractor יחד עם ההפרש מהחציון. ניתן לראות שאין כמעט רעשים אך חלקים רבים מגוף הדמות חתוכים



שימוש ב KNN subtractor יחד עם טשטוש חציוני. ניתן לראות שהדמות שלמה אך מתווסף רעש מהרקע



**Matting**

את בלוק ה-matting נחלק לשני שלבים, יצירת הסרטון matted, ויצירת האלפא "הלא מיוצב".

יצירת Matting:

בשלב זה אנו מעוניינים להפיק תחילה סרטון "אלפא" סרטון בעל ערכים מ-0 עד 1, כאשר 0 מייצג אזור בו אנו בטוחים שהוא הרקע, 1, מייצג אזור שבו אנו בטוחים שהדמות צריכה להימצא, וערכי ביניים המייצג שילוב של השניים (undecided).  
כדי לחשב את סרטון האלפא, השתמשנו בתהליך המתואר במצגת "Implementation of Video Colorization and Matting".

בשלב זה עברנו לעבודה בערוצי צבע HSV כשעבדנו ספציפית עם ערוץ ה-Value.  
רצינו לחשב את הסבירות להשתייך לרקע ושל הדמות. לצורך כך השתמשנו בפונקציה המחשבת KDE (Kernel Density Estimation) שראינו בתרגולים.

התלבטנו אם להשתמש בסקריבלים מהמשתמש או בתמונה הבינארית עצמה כדי לבחור את האזור אותו מכניסים כקלט ל-KDE. השיקולים לשני אלו היו שעבור סקריבלים מהמשתמש ישנה אי וודאות מסוימת לגבי הסקריבל שייבחר וכך גם על התוצאה אך אנחנו מניחים שהאזור שיסומן יהיו הרקע/הדמות באופן וודאי. מנגד בשימוש במסכה הבינארית אין תלות במשתמש אך ישנם פריימים בהם התמונה אינה מושלמת ויש אזורים מוסתרים (בהמשך נבין שהקלט הנ"ל משמעותי מאוד). בסופו של דבר בחרנו בשימוש במסכה הבינארית כseed.

את המידע על הדמות לקחנו מתוך סרט הקלט המיוצב, אשר עליו אנו מלבישים את המסכה ואת המידע על הרקע, חישבנו בעזרת החלקת חציון על הסרטון (במקרה זה מכיוון שאנו מעוניינים הערכי הפיקסלים ולא בסדר שלהם על המטריצה לא הייתה בעיה להשתמש גם בחציון רק על 3 הפריימים הראשונים והאחרונים של הסרטון). תמונת רקע זאת נשמרת גם בתיקיה Temp (אולם ניתן להריץ את התוכנית גם בלעדיה).  
כמו כן, כדי לקצר את זמן החישוב, ביצענו resize על תמונת הרקע.

בעזרת התוצאות של פונקציית ה KDE וכלל בייס, חישבנו את פונקציות ההסתברות של הרקע ושל הדמות. באלו השתמשנו כדי ליצור מפת הסתברויות לכל התמונה.

בשלב השני אנו רוצים לחשב את המרחק הגיאודזי. לצורך כך השתמשנו בספריה GeodisTK, בפונקציה מסוג raster Scan. בתור seed עבור הפונקציה בחרנו להשתמש בתמונה הבינארית שלנו בכל פריים, לאחר ביצוע של טרנספורמציה מורפולוגית מסוג erosion (הקטנה של החלק הלבן בתמונה) בחישוב הדמות ו-dilation לחישוב הרקע עשינו זאת כדי לתת כ-seed רק את האזור שבו אנו בטוחים שנמצאים הדמות/הרקע.

מכיוון שהחישוב של המרחק הגיאודזי הינו כבד מאוד, בכל פריים, לא העברנו אליה כקלט את כל התמונה, אלא ביצענו tracking מסוג csrt (הסבר יותר נרחב עליו אפשר למצוא בחלק ה tracking). והעברנו לפונקציה מידע רק על האזור שאחריו אנו עוקבים.

לאחר מכן, בעזרת התוצאה של המרחק הגיאודזי, ופונקציית ההסתברות, חישבנו את מפת האלפא בעזרת הנוסחאות הבאות:

ניסינו ערכים שונים, כשהעיקריים ביניהם היו כשבסוף בחרנו ב-

בשלב זה יש בידינו את מפת האלפא, אולם אנו מעוניינים להתחשב בה רק בגבול הדמות והרקע. בנינו לשם כך מסכת Trimap בה האזורים הוודאיים למיקום הדמות (אלו שקיבלנו מה-BS) היו 1, אלו של הרקע היו 0 והאזורים בשולי הדמות הם Unknown region. את האזור הזה יצרנו באמצעות ביצוע dilation, וביצוע נפרד של erosion עקב העובדה שהניחוש הראשוני מה-BS כלל מעט מהרקע בגבולות הדמות. לבסוף יצרנו תמונת אלפא אשר מורכבת ממפת האלפא ב-Unknown region, ומהתמונה הבינארית בכל מקום אחר.   
לאחר שקיבלנו את פריים האלפא השתמשנו בנוסחה הבאה בשביל לייצר את matted:

*יצירת ה unstabilized alpha:*

*עבור שלב זה, אנו מבצעים תהליך הדומה מאוד לייצוב, מחשבים את המטריצה m כפי שהוסבר לעיל, מחשבים עבורה צורה הופכית, ומבצעים warp על סרטון האלפא (המיוצב) כדי לקבל סרטון לא מיוצב.*

**עקיבה – Tracking**

בחלק זה לצורך יעילות החלטנו להשתמש בtracker של opencv.

השתמשנו בסוג CSRT שנתן תוצאות טובות יחסית על הקלט עם הרקע החדש (הסרטון matted).

בנוסף השתמשנו בעוקב הזה גם בבלוק ה-Matting כדי לחשב את המרחק הגאודזי על חלק קטן של התמונה המכיל את הדמות ולא על התמונה כולה כדי לחסוך זמן.