נושא:

Fast image dehazing using guided joint bilateral filter

Recent Advances in Image Dehazing

מרצה:

אמיר הנדלמן

מגיש:   
רוסלן אוסמנוב - 327480026

[Ruslan31bar@gmail.com](mailto:Ruslan31bar@gmail.com)

Git : [LINK](https://github.com/deamon312/image-processing/tree/main/Project%202)

[GUI Fast image Dehazin](https://drive.google.com/file/d/1dIOsefgTGr2eBJEAxaHKWFaTjAHgUwLR/view?usp=sharing)g (ראה אופן שימוש עמ' 12)

[GUI Homomorphic Dehazing](https://drive.google.com/file/d/1uShjdk7MF0RysBE9Hi04XCihFhrk_wfA/view?usp=sharing) (ראה אופן שימוש עמ' 18)

**תוכן עניינים**

Contents

[Fast image dehazing using guided joint bilateral filter. 3](#_Toc136082827)

[מבוא 3](#_Toc136082828)

[רקע כללי 3](#_Toc136082829)

[הסרת ערפל על בסיס סינון 4](#_Toc136082830)

[אלגוריתם 5](#_Toc136082831)

[Guided joint bilateral filter. 5](#_Toc136082832)

[הערכת אור אטמוספירה 7](#_Toc136082833)

[Scene radiance recovering 7](#_Toc136082834)

[Adaptive Histogram Equalization (תוספת שלי) 7](#_Toc136082835)

[תוצאות 8](#_Toc136082836)

[GUI -אופן שימוש 12](#_Toc136082837)

[Recent Advances in Image Dehazing 13](#_Toc136082838)

[מבוא 13](#_Toc136082839)

[Homomorphic filtering 13](#_Toc136082840)

[תוצאות השוותיות בין שני שיטות לניקוי אובך: 16](#_Toc136082841)

[מסקנות: 18](#_Toc136082842)

[GUI -אופן שימוש 18](#_Toc136082843)

[Appendix A 19](#_Toc136082844)

[Appendix B 28](#_Toc136082845)

# Fast image dehazing using guided joint bilateral filter.

## מבוא

תופעת אובך או ערפל הנגרמת על ידי חלקיקים מרחפים עדינים באוויר, מה שמפחית את הראות ביישומי ראייה ממוחשבת. שיטות שונות הוצעו להסרת אובך מתמונות, אך עדיין נותרו אתגרים בגלל אופי הבעיה. שיטות מסוימות מחלצות מידע עומק ממספר תמונות או משתמשות במידע נוסף, אך הן אינן מתאימות ליישומים בזמן אמת. מחקרים אחרונים התמקדו בהסרת אובך מתמונה בודדת, תוך שימוש בטכניקות כמו מיקסום הניגודיות המקומית, הערכת זוהר הסצנה ויישום ה-Dark Channel Prior (DCP). עם זאת, לשיטות אלו עשויות להיות מגבלות, כגון הפקת צבעים רוויים מדי או דרישה לתהליכים אינטנסיביים מבחינה חישובית. כדי להתמודד עם חששות היעילות, חוקרים הציעו טכניקות שחרור מואצות, כגון מסננים מונחים ואלגוריתמים מהירים המשתמשים במסננים חציוניים. בשיטה המוצגת, מוצג אלגוריתם חדש להסרת ערפל בתמונה אחת, אשר משחזר צעיף אווירה מדויק יותר על ידי מינוף מידע עומק הסצנה. האלגוריתם משתמש במידע קצה מתמונת הקלט כדי לסנן את צעיף האטמוספרה הראשוני, וכתוצאה מכך תוצאות משופרות של חוסר ערוב כאשר העומק משתנה באופן פתאומי.

## רקע כללי

בראייה ממוחשבת, היווצרות תמונות אובך מתוארת בדרך כלל על ידי תהליך הנחתת האטמוספרה:



כאשר I (x) הוא תמונת האובך, J(x) התמונה נטולת אובך, A הוא אור האטמוספירה הכולל, ו-t (x) הוא תווך, כאשר האטמוספירה הומוגנית, t(x) = exp(−βd(x)). כאן, β הוא מקדם הפיזור של האטמוספירה, ו-d הוא עומק הסצנה. המטרה של הסרת אובך היא לשחזר את J(x), A ו-t(x) מ-I(x). המונח J(x)t(x) הוא הנחתה ישירה, מה שמצביע על כך שעורפל יגרום לזוהר הסצנה להיחלש באופן אקספוננציאלי עם עומק הסצנה d בתווך. המונח A(1 − t(x)) הוא הצעיף האטמוספרי (פיזור אור אטמוספרי), הגורם לטשטוש, שינוי צבע ועיוות בסצנה.

שיפור נראות התמונה על ידי מיקסום הניגודיות של התמונה המתקבלת, וניסוח מחדש של הבעיה כמיקסום צעיף האווירה V(x) = A(1−t(x)) בהנחה ש-V(x) חלק רוב הזמן, למעט לאורך הקצוות עם קפיצות עומק גדולות. מכיוון שפונקציית האופטימיזציה הייתה מסובכת מבחינה חישובית, הציעו אלגוריתם שחזור נראות מהיר על ידי שימוש בגישת סינון לחישוב צעיף האטמוספירה V(x). הם הניחו ש-V(x) רצוי צריך לעמוד בשני האילוצים הבאים: (1) הערך V(x) חיובי (V(x) ≥ 0) בכל פיקסל; (2) הערך של V(x) אינו גבוה מה- רכיבי

של I (x), V(x) ≤ W(x), כאשר הסימון W(x) = minc∈{r,g,b}(Ic(x)) הוא רכיבי הצבע המינימליים של I(x) ).

עם שני האילוצים והתצפיות הללו, השתמשו בסינון חציוני כדי להניב את הפונקציה הרצויה V(x). תחילה הם סיננו את ה-W(x) באמצעות מסנן חציוני כדי לקבל B(x) ולהקל על ההשפעה של מרקם מנוגד להסרת הערפל, הם גם יישמו את ההבדל של הממוצע המקומי B(x) וסטיית התקן המקומית של W(x). לבסוף, הם הכפילו את C(x) בגורם קנה מידה [0, 1] p כדי לשלוט בעוצמת שחזור הראות. פיזור האור באטמוספירה V(x) חושב לפי השלבים הבאים:

A picture containing text, font, white, typography

Description automatically generated

כאשר Ω הוא חלון מרובע של מסנן חציוני. מכיוון שהמסנן החציוני עצמו אינו משמר קצוות ואינו קונפורמי, מידע רב על הקצה בתמונה V (x) המתקבלת אובד לאחר סינון חציוני פעמיים, הכולל את הקצוות שבהם ערך העומק משתנה באופן פתאומי. יחד עם זאת, קפיצה גדולה בצעיף האטמוספרה חשובה מאוד לשחזור תמונה. אם מידע הקצה חסר, האלגוריתם לא יכול לזהות את האובך במיקומים אלה, וכתוצאה מכך הסרת אובך לא מלאה. את מוצא של W(x)ואת V(x) ניתן לראות באיור הבא:

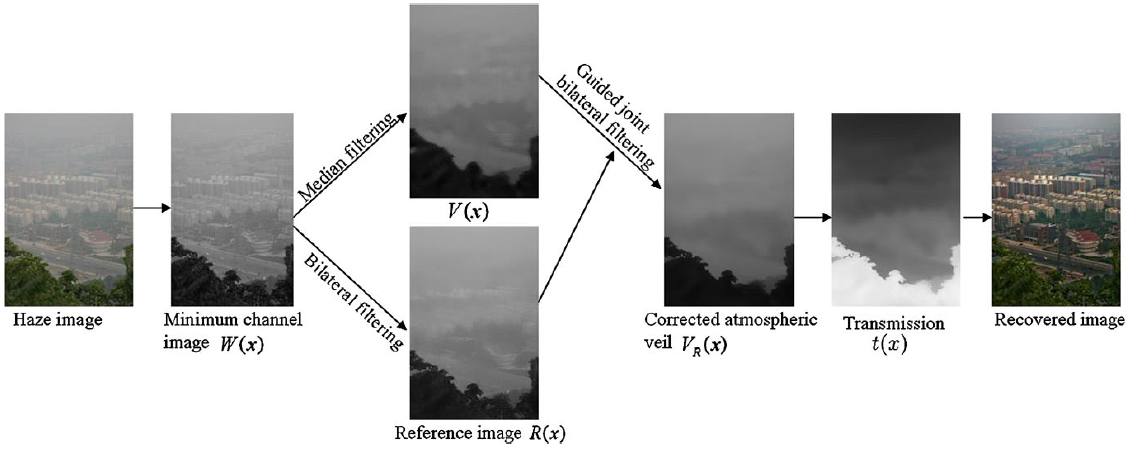
|  |  |  |
| --- | --- | --- |
|  |  |  |

איור 1

## הסרת ערפל על בסיס סינון

אנו רואים שהשונות של צעיף האטמוספירה V(x) = A(1−t(x)) תלויה בעיקר בעומק d של הסצנה, כלומר במרחק של האובייקטים לצופה. לפיכך, צעיף האטמוספרה הרצוי צריך להיות חלק ועוצמתו צריכה להשתנות בהדרגה בהתאם לעומק הסצנה, ובאזורים בעלי אותו עומק צריכה להיות לעוצמת ה-V(x) ערך דומה. כאשר אנו מסיקים את ה-V(x) מה-W(x), נותר מידע מרקם רב כמו מידע קצה ב-W(x). מידע הקצה חשוב מכיוון שהם מניחים קפיצות עומק גדולות אפשריות בין העצמים, כגון הקצוות בין העלים והקיר לפיכך, יש צורך לשחזר מידע קצה. אנו חושבים ששיטת סינון אחת מתאימה צריכה לשמור על צעיף האווירה חלק, לשמר את קפיצות העומק ולהסיר את מידע המרקם המיותר. כדי להגיע למטרה לעיל, אנו מנסים לתקן את צעיף האטמוספירה V(x). מטרת אופרטור תיקון זה היא לשחזר את מידע הקצה ב-V(x), ולהקטין את פרטי הטקסטורה ב-V(x), מכיוון שמידע מרקם אינו מרמז על שינויים בעומק, אנו לוקחים תמונה המכילה את מידע הקצה של תמונת המקור כתמונת הייחוס, ומציעים מסנן מונחה משותף דו-צדדי כדי לשפר את צעיף האטמוספירה V(x) שהושג באמצעות סינון חציוני, כפי שמוצג באיור 1.

### אלגוריתם



איור 2 תרשים הזרימה של אלגוריתם

### Guided joint bilateral filter.

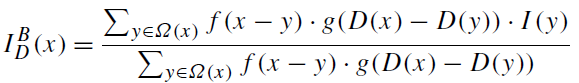
המסנן הדו-צדדי, שהוצג במקור, נמצא בשימוש נרחב בקהילות גרפיקה ממוחשבת וראייה ממוחשבת, המסנן הדו-צדדי מחשב את פלט המסנן בפיקסל כממוצע משוקלל של פיקסלים שכנים והוא מסוגל לשמר את הקצוות של תמונות מעובדות. עבור כל פיקסל בתמונה I, תן ל-Ω(x) להיות התיקון המקומי שמרכזו ב-,x I(x) ו-I(y) יהיו ערך העוצמה המתאים של פיקסלים x ו-y, ואז ערך העוצמה המסוננת של x הינה:

A picture containing text, font, line, white

Description automatically generated

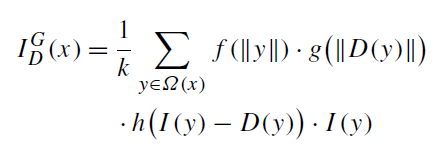
כאשר f ו-g הם גרעיני המסנן המרחביים והטווח, בהתאמה.

המסנן הדו-צדדי מוכלל לסינון הדו-צדדי המשותף, שבו גרעין הטווח מחושב על סמך תמונת הדרכה אחרת D, ואז ערך העוצמה המסוננת של x הינה:



לפיכך, ה- של התמונה המסוננת תקבל את מידע הקצה של תמונת הייחוס D. המסנן הדו-צדדי המשותף מועדף במיוחד כאשר תמונת הקלט אינה אמינה לספק מידע קצה, למשל, כאשר היא רועשת מאוד או היא תוצאת ביניים בעיבוד תמונה .

על מנת לשמר את הקצה ולתקן את פרטי הקצה הלא מושלמים, אנחנו לא רק צריכים לשקול את ההבדל בתמונת ההפניה, אלא גם לדרוש לשקול עוד יותר את ההבדל בין התמונה שיש לסנן לבין תמונת ההפניה, לתת משקלים גדולים יותר לפיקסלים עם סטייה קטנה יותר ומשקולות קטנות יותר לפיקסלים עם סטייה גדולה יותר. באמצעות פעולת הסינון הזו, נוכל לתקן את התמונה לכיוון תמונת ההפניה. מסנן זה נקרא מסנן משותף דו-צדדי מודרך, וערך העוצמה המסוננת של x הוא:



כאשר ,



לאחר חישוב צעיף האטמוספירה V(x), שיטה זו מאבדת מידע רב. אנו מנסים להוסיף מידע קצה לתמונה המקורית של I(x) עד V(x), ובינתיים מצמצמים את פרטי המרקם של V(x). מכיוון שמפת ערוץ הבהירות המינימלית W(x) מכילה את תכונות הקצה ופרטי המרקם של תמונת הקלט, אנו מציעים להשתמש ב-W(x) כרמז לשיפור מידע הקצה של V(x) סביב האזורים עם קפיצות עומק פתאומיות . אנו משתמשים תחילה במסנן דו-צדדי ב-W(x) כדי לסנן כמה פרטי מרקם, בעוד שניתן לשמר היטב את תכונות הקצה:

A picture containing text, font, line, white

Description automatically generated

A picture containing font, text, white, calligraphy

Description automatically generatedלאחר מכן ניקח את התמונה המסוננת R(x) כתמונת התייחסות לסינון צעיף האטמוספירה V(x) באמצעות מסנן המונחה המשותף:

כאשר k הוא הגורם המנרמל, ליבת מסנן הטווח:



|  |  |  |  |
| --- | --- | --- | --- |
| Kernel = 21 , sigma\_r=40 , sigma\_t=45 | | Kernel = 15 , sigma\_r= 9 , sigma\_t=9 | |
|  |  |  |  |

איור 3 . השפעת גודל גרעין וסטיות תקן של גאוסיין

### הערכת אור אטמוספירה

אור האטמוספירה הכוללת A מוערך בדרך כלל מפיקסלים עם האובך הצפוף ביותר, לדוגמה, לוקחים את ערך הבהירות הגבוה ביותר של התמונה כאור הסביבה. אבל הפיקסל הבהיר ביותר עשוי להיות האובייקטים הלבנים. אנו משתמשים בערוץ האפל (W) כדי לשפר את הדיוק של אור האטמוספירה: ראשית, בחירת הפיקסלים הבהירים ביותר (0.2%) בערוץ האפל, המתאימים לאזורים המעורפלים ביותר; לאחר מכן, בחירת הפיקסל הבהיר ביותר מבין הפיקסלים באותו מיקום בתמונת הקלט המעורפלת כמו אור האווירה הכללי A.

### Scene radiance recovering

בהינתן V\_R(x) ואור האטמוספירה הכללי A, ניתן לקבל את t (x) על ידי: t(x) = 1 − ωV\_R(x)/A, כאשר הפרמטר ω ∈ (0, 1] משמש לשימור מעט אובך בסצנה המרוחקת והופך את התמונה המשוחזרת לטבעית יותר. ברוב הניסויים זה, הערך של ω הוא 0.95. ניתן לחשב את הקרנת הסצנה הסופית (התמונה הברורה נטולת האובך) כ- כאשר t0 הוא הגבול התחתון של השידור, המשמש כדי למנוע רעשים באזורי אובך צפופים מאוד.

### Adaptive Histogram Equalization ( \*\*תוספת שלי)

השוואת היסטוגרמה אדפטיבית היא שיטה המשמשת לשיפור הניגודיות של תמונה על ידי חלוקה מחדש של עוצמות הפיקסלים. בעוד השוואת היסטוגרמה מסורתית פועלת על כל התמונה, מחלקת את התמונה לאזורים קטנים יותר ומבצעת השוואת היסטוגרמה באופן עצמאי בכל אזור. זה מאפשר שיפור ניגודיות טוב יותר באזורים מקומיים.

* המרת תמונת הצבע מ-RGB למרחב צבע LAB. מרחב הצבעים של LAB מפריד בין רכיב הבהירות (L), המייצג את הבהירות, לבין מידע הצבע (רכיבי a ו-b).
* השוואת היסטוגרמה על ערוץ L. שלב זה משפר את הניגודיות ברכיב הבהירות.
* שילוב ערוץ L המתוקן עם מידע הצבע המקורי בערוצי a ו-b כדי לקבל את תמונת המעבדה המתוקנת.
* המרת תמונת LAB מתוקנת חזרה למרחב הצבע RGB.

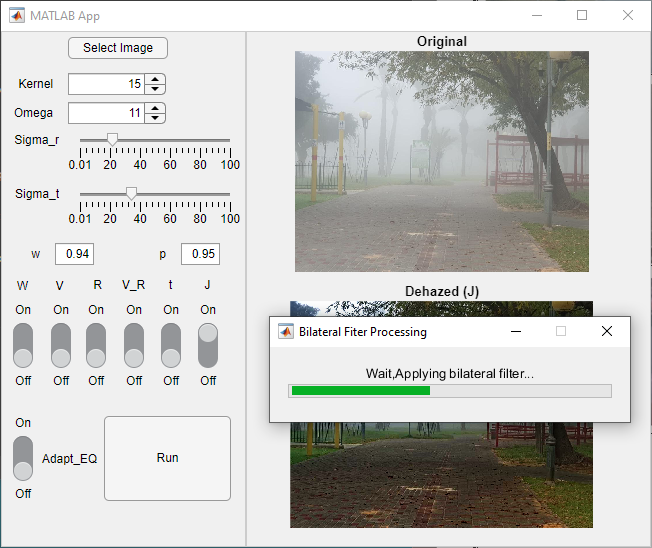
|  |  |  |
| --- | --- | --- |
|  |  |  |

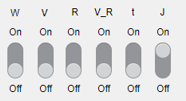
## תוצאות

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## GUI -אופן שימוש

GUI רשום בשפת מטלב



* Select Image– בחירה תמונה מתוך התיקיה.
* Kernel – גודל של גרעין עבור מסנן בילטרלי (R, V\_R)
* Omega – גודל של גרעין טשטוש במסנן חציוני (V)
* Sigma\_r, Sigma\_t – סטית תקן של גרעני גאוסיאני (R,V\_R)
* p – משקל כדי לשלוט בעוצמת שחזור הראות
* w - משקל לשימור מעט אובך בסצנה המרוחקת
* Run – כפתור הרצה של אלגוריתם.
* בוררים לצורך הצגה של תמונה

בשלבי עיבוד שונים

* **הקוד נמצא ב Appendix A - Script ו** [**Git**](https://github.com/deamon312/image-processing/tree/main/Project%202/First%20Article)

# Recent Advances in Image Dehazing

## מבוא

בתנאי מזג אוויר כמו ערפל ואובך, איכות התמונות יורדת מאוד עקב השפעת החלקיקים באטמוספרה. חלקיקים מרחפים יפזרו אור ויגרמו להפחתת האור המוחזר מהסצנה והאור האטמוספרי המפוזר יתערבב גם עם האור המתקבל במצלמה וישנה את הניגודיות והצבע של התמונה. לכן, יש צורך במערכות ראייה ממוחשבת כדי לשפר את האפקטים החזותיים של התמונה ולהדגיש את תכונות התמונה. טכניקת הסרת אובך של תמונה, הידועה גם בשם "הסרת אובך" או "הסרת ערפל" היא רק הטכניקה להפחתת או אפילו להסיר הפרעות עקב אובך על ידי גישות מיוחדות, על מנת להשיג אפקטים חזותיים מספקים ולקבל מידע שימושי יותר. בתיאוריה, ניקוי אובך של תמונה מסיר אפקטים חזותיים לא רצויים ולעתים קרובות נחשב כטכניקה לשיפור תמונה. עם זאת, היא נבדלת משיטות מסורתיות להסרת רעשים ושיפור ניגודיות שכן השפלה של פיקסלים תמונה שנגרמה על ידי נוכחות אובך תלויה במרחק בין האובייקט להתקן הרכישה ובצפיפות האזורית של האובך. ההשפעה של ערפול על פיקסלים של תמונה גם מדכאת את הטווח הדינמי של הצבעים.

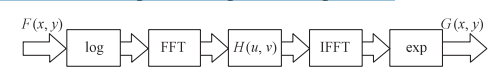
A picture containing text, font, screenshot, parallel

Description automatically generatedבהתבסס על הבדלים בעקרונות שחרור מערפל, ניתן לחלק את השיטות הנוכחיות לשלוש קטגוריות: שיטות מבוססות שיפור תמונה, שיטות מבוססות היתוך תמונה ושיטות מבוססות שחזור תמונה. שיטות המבוססות על שיפור תמונה אינן לוקחות בחשבון את הגורם להידרדרות התמונה, אלא משתמשות בעיקר בשיטות עיבוד תמונה ממוקדות לשיפור הניגודיות והפרטים, ולשיפור האפקטים החזותיים של התמונה. שיטות מבוססות היתוך תמונה ממקסמות את המידע המועיל מערוצי מקור מרובים כדי ליצור תמונה באיכות גבוהה, ללא צורך במודל פיזי. שיטות מבוססות שחזור תמונה מבססות מודל השפלה של תמונה מעורפלת על ידי לימוד המנגנונים הפיזיים של הדמיה אופטית, היפוך תהליכי השפלה ופיצוי על עיוות הנגרם מתהליכי השפלה אלו על מנת לקבל תמונות ברורות ללא אובך.

## Homomorphic filtering

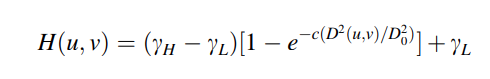
בתנאי ערפל, רכיבי התדר הנמוך של תמונה משופרים, כך שניתן להשתמש במסנן מעביר גבוהים לסינון תמונות כדי לדכאה תדרים נמוכים ולשפר תדרים גבוהים. שיפור תחום התדר משתמש תמיד בניתוח פורייה ובשיטות אחרות כדי להמיר תמונה לתחום התדר. לאחר השלמת פעולת הסינון, מתבצעת טרנספורמציה הפוכה חזרה לתחום המרחבי. שיטות טיפוסיות המבוססות על תחום התדר כוללות סינון הומו-מורפי, והתמרת wavelets.

העיקרון של סינון הומו-מורפי הוא חלוקת התמונה לרכיב קרינה ורכיב השתקפות. מרכיב הקרינה של התמונה המעורפלת מאופיין בשונות איטית במרחב, ומרכיב ההשתקפות מקושר לרוב לפרטי הסצנה. שיפור התמונה מושג על ידי הסרת רכיב הקרינה. על ידי שילוב של סינון תדרים עם טרנספורמציה בתחום האפור, ניתן להשתמש בטווח הדינמי של התמונה הדחוסה כדי לשפר את איכות התמונה. לפיכך, העיקרון הבסיסי של סינון הומו-מורפי לניקוי אובך עדיין מבוסס על מודל ההארה. תרשים הזרימה של אלגוריתם זה מוצג באיור 1. כאשר logהינה התמרה הלוגריתמית, FFT הוא התמרת פורייה, H(u, v) היא פונקציית סינון התדרים, IFFT היא התמרת פורייה הפוכה ו-exp היא הפעולה הופכית ל log .

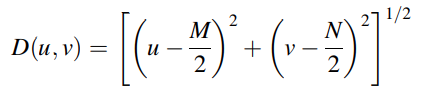


איור 1

אלגוריתם הסינון הומו-מורפי יכול להסיר אזורים לא אחידים שנוצרו על ידי אור תוך שמירה על מידע קווי המתאר של התמונה. עם זאת, הוא זקוק לשתי טרנספורמציות פורייה, פעולה אקספוננציאלית אחת ופעולה לוגריתמית אחת לכל פיקסל של התמונה.

H(u.v) מייצג פונקציית תמסורת מסנן ההומו-מורפי, (גאוס):

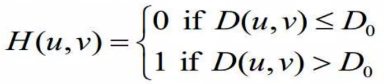
כאשר D0 הוא תדר הקטעון של המסנן, ו- הם ערכי המינימום והמקסימום של המסנן, וקבוע c משמש בעיקר כדי לשלוט על ההטיה של פונקציית העברת המסנן. D(u,v) הוא המרחק מנקודה (u,v) למקור התדר, המוגדר כדלקמן.

****

A picture containing text, font, line, white

Description automatically generatedבנוסף הוספתי אופציה למסנן BUTTERWORTH ומסנן אידיאלי :

מסנן מעביר גבוהים של Butterworth שומר על תדרים מחוץ לרדיוס D0. יש לו מעבר הדרגתי מ-0 ל-1 כדי להפחית חפצי צלצול. מסנן (BHPF) של Butterworth בסדר n ותדר חיתוך D0 .

****מסנן מעביר גבוהים אידיאלי

|  |  |  |
| --- | --- | --- |
| Ideal High-Pass | Butterworth High-Pass | Gausian High-Pass |
|  |  |  |
|  |  |  |

נציג את מוצא לאחר כל שלב באלגוריתם שבאיור 1 בנוסף הוספתי אוםציה לתיקון היסטוגרמה אדפטיבית כמו בחלק הראשון של הפרויקט זה (נציג רק בעזרת מסנן גאוסיאני):

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

מסקנות:

היתרון של שיטה זו היא ביקר זמן חישוב נמוך מאוד ופשטות של השיטה ,אך ככל שמגדלים את רדיוס של מסנן כך יורדת גם הבהירות ככלית של התמונה.

## תוצאות השוותיות בין שני שיטות לניקוי אובך:

|  |  |  |
| --- | --- | --- |
| Homomorphic Dehazing | Bilateral Dehazing | Original |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### מסקנות:

לסיכום,המסנן הדו-צדדי המשותף והפילטר ההומומורפי לדי-ערפול של תמונות מציע פתרון מבטיח שהמתמודד ביעילות עם האתגרים שמציבים תנאים מעורפלים. גישה זו לוקחת בחשבון הן את יעילות זמן הריצה והן את השמירה על תכונות חיוניות כגון חדות, דיוק צבע ואיכות חזותית כוללת.

גישה עם המסנן הדו-צדדי המשותף, על ידי התחשבות בדמיון במרחב ובטווח, מפחית בהצלחה את האובך תוך שמירה על פרטים חשובים בתמונה. היכולת שלו לשמור על חדות הקצה והמרקם הופכת אותו לבחירה אידיאלית למשימות של ניקוי אובך. יתר על כן, מסנן זה מבטיח שזמן הריצה יישאר יעיל, ומאפשר עיבוד כמעט בזמן אמת של תמונות מעורפלות.

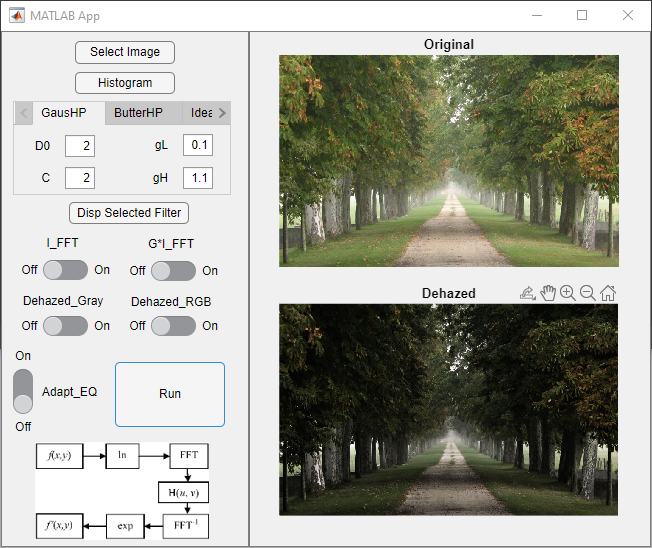
גישה עם הפילטר ההומורפי, לעומת זאת, יעיל במיוחד בשיפור הניגודיות בתמונות מעורפלות. על ידי פירוק התמונה לרכיבים התדר, זה מקל על הסרת אובך ובו זמנית משפר את הנראות של פרטים עדינים. מסנן זה מבטיח שזמן הריצה יישאר יעיל, ומאפשר עיבוד בזמן אמת של תמונות מעורפלות.

ההבדל הוא בעזרת גישה עם מסנן הדו-צדדי משותף ניתן לשחזר ולשמור על צבעי של התמונה

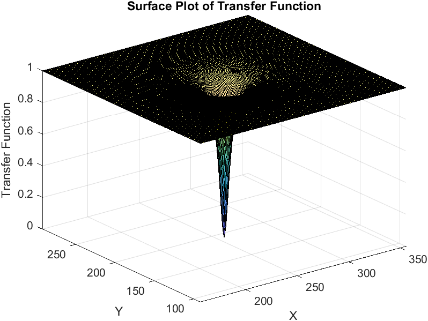
## GUI -אופן שימוש

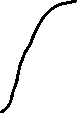
GUI רשום בשפת מטלב

|  |
| --- |
| **A screenshot of a computer  Description automatically generated with medium confidence** |
| **A screenshot of a computer program  Description automatically generated with medium confidence** |
| **A screenshot of a computer  Description automatically generated with medium confidence** |

****



****



* Select Image – בחירת תמונה
* Histogram – הצגת היסטוגרמה של תמונת אפור עבור תמונת מקור
* **A screenshot of a computer

  Description automatically generated with medium confidence** - בחירת פילטר מעביר גבוהים
* Display Selected Filter – הצגת פונק תמסורת של מסנן הנבחר.
* A picture containing text, screenshot, font, design

  Description automatically generated - בחירת אופציה להצגת תמונות בכל שלב
* Run – כפתור הרצה של אלגוריתם.
* **הקוד נמצא ב Appendix B - Script ו** [**Git**](https://github.com/deamon312/image-processing/tree/main/Project%202/Second%20Article)

# Appendix A

classdef FastDehazeImage\_exported < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

GridLayout matlab.ui.container.GridLayout

LeftPanel matlab.ui.container.Panel

SelectImageButton matlab.ui.control.Button

KernelSpinnerLabel matlab.ui.control.Label

KernelSpinner matlab.ui.control.Spinner

OmegaSpinnerLabel matlab.ui.control.Label

OmegaSpinner matlab.ui.control.Spinner

wEditFieldLabel matlab.ui.control.Label

wEditField matlab.ui.control.NumericEditField

pEditFieldLabel matlab.ui.control.Label

pEditField matlab.ui.control.NumericEditField

WSwitchLabel matlab.ui.control.Label

WSwitch matlab.ui.control.Switch

VSwitchLabel matlab.ui.control.Label

VSwitch matlab.ui.control.Switch

V\_RSwitchLabel matlab.ui.control.Label

V\_RSwitch matlab.ui.control.Switch

RunButton matlab.ui.control.Button

Sigma\_tSliderLabel matlab.ui.control.Label

Sigma\_t matlab.ui.control.Slider

Sigma\_rSlider\_2Label matlab.ui.control.Label

Sigma\_r matlab.ui.control.Slider

RSwitchLabel matlab.ui.control.Label

RSwitch matlab.ui.control.Switch

tSwitchLabel matlab.ui.control.Label

tSwitch matlab.ui.control.Switch

Adapt\_EQSwitchLabel matlab.ui.control.Label

Adapt\_EQSwitch matlab.ui.control.Switch

JSwitchLabel matlab.ui.control.Label

JSwitch matlab.ui.control.Switch

RightPanel matlab.ui.container.Panel

UIAxes\_2 matlab.ui.control.UIAxes

UIAxes matlab.ui.control.UIAxes

end

% Properties that correspond to apps with auto-reflow

properties (Access = private)

onePanelWidth = 576;

end

properties (Access = private)

imageData = []

imageW = []

imageV = []

imageV\_R = []

imageR = []

imaget = []

imageJ = []

imageJ\_adapt = []

end

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: SelectImageButton

function SelectImageButtonPushed(app, event)

close all;

[file ,path] = uigetfile({'\*.jpg;\*.jpeg;\*.png;\*.gif;\*.tif';'\*.\*'},'File Selector');

selectedfile = fullfile(path,file);

% Check if a file was selected

if isequal(file, 0) % No file selected

msgbox('There is NO image selected');

else

% Read the image file

app.imageData=imread(selectedfile);

imshow(app.imageData, 'Parent', app.UIAxes);

[height, width, ~] = size(app.imageData);

app.UIAxes.XLim = [0 width];

app.UIAxes.YLim = [0 height];

app.KernelSpinner.Enable='on';app.KernelSpinnerLabel.Enable='on';

app.OmegaSpinner.Enable='on';app.OmegaSpinnerLabel.Enable='on';

app.Sigma\_r.Enable='on';app.Sigma\_rSlider\_2Label.Enable='on';

app.Sigma\_t.Enable='on';app.Sigma\_tSliderLabel.Enable='on';

app.pEditField.Enable='on';app.pEditFieldLabel.Enable='on';

app.wEditField.Enable='on';app.wEditFieldLabel.Enable='on';

app.tSwitch.Enable='on';app.tSwitchLabel.Enable='on';

app.V\_RSwitch.Enable='on';app.V\_RSwitchLabel.Enable='on';

app.RSwitch.Enable='on';app.RSwitchLabel.Enable='on';

app.VSwitch.Enable='on';app.VSwitchLabel.Enable='on';

app.WSwitch.Enable='on';app.WSwitchLabel.Enable='on';

app.JSwitch.Enable='on';app.JSwitchLabel.Enable='on';

app.Adapt\_EQSwitch.Enable='on';app.Adapt\_EQSwitchLabel.Enable='on';

app.RunButton.Enable='on';

end

end

% Button pushed function: RunButton

function RunButtonPushed(app, event)

close all;

[height, width, ~] = size(app.imageData);

sigma\_s=0.03\*min(height,width);

sigma\_r=app.Sigma\_r.Value;

sigma\_t=app.Sigma\_t.Value;

p=app.pEditField.Value;

w=app.wEditField.Value;

t0=0.3;

radius = app.KernelSpinner.Value;

omega(1:2) = app.OmegaSpinner.Value;

app.imageW = double(min(app.imageData,[],3));

if strcmp(app.WSwitch.Value,'On')

figure,imshow(uint8(app.imageW)),title('W');

end

B = medfilt2(app.imageW,omega,'symmetric');

C=B-medfilt2(abs(app.imageW-B),omega,'symmetric');

app.imageV=max(min(p.\*C,app.imageW),0);

if strcmp(app.VSwitch.Value,'On')

figure,imshow(uint8(app.imageV)),title('V');

end

app.imageR=bilat\_filter(app.imageW,radius,sigma\_s,sigma\_r);

if strcmp(app.RSwitch.Value,'On')

figure,imshow(uint8(app.imageR)),title('R');

end

app.imageV\_R=bilat\_filter\_joint(app.imageV,app.imageR,radius,sigma\_s,sigma\_r,sigma\_t);

if strcmp(app.V\_RSwitch.Value,'On')

figure,imshow(uint8(app.imageV\_R)),title('V\_{R}');

end

A = min([estimateAtmosphericLight(app.imageW), max(max(255-app.imageW))]);

app.imaget=ones(height,width)-w\*app.imageV\_R/A;

if strcmp(app.tSwitch.Value,'On')

figure,imshow(app.imaget),title('depth image t');

end

image\_double=double(app.imageData);

app.imageJ=zeros(size(app.imageData));

app.imageJ(:,:,1)=(image\_double(:,:,1)-A)./max(app.imaget,t0)+A;

app.imageJ(:,:,2)=(image\_double(:,:,2)-A)./max(app.imaget,t0)+A;

app.imageJ(:,:,3)=(image\_double(:,:,3)-A)./max(app.imaget,t0)+A;

imshow(uint8(app.imageJ), 'Parent', app.UIAxes\_2);

[height, width, ~] = size(app.imageJ);

app.UIAxes\_2.XLim = [0 width];

app.UIAxes\_2.YLim = [0 height];

if strcmp(app.JSwitch.Value,'On')

figure,imshow(uint8(app.imageJ)),title('J');

figure,imshow(uint8(app.imageData)),title('Original');

end

if strcmp(app.Adapt\_EQSwitch.Value,'On')

LAB = rgb2lab(uint8(app.imageJ));

L = LAB(:,:,1)/100;

L = adapthisteq(L,'NumTiles',[8 8],'ClipLimit',0.005);

LAB(:,:,1) = L\*100;

app.imageJ\_adapt = lab2rgb(LAB);

figure,imshow(app.imageJ\_adapt),title('J\_{adaptEQ}');

end

end

% Changes arrangement of the app based on UIFigure width

function updateAppLayout(app, event)

currentFigureWidth = app.UIFigure.Position(3);

if(currentFigureWidth <= app.onePanelWidth)

% Change to a 2x1 grid

app.GridLayout.RowHeight = {516, 516};

app.GridLayout.ColumnWidth = {'1x'};

app.RightPanel.Layout.Row = 2;

app.RightPanel.Layout.Column = 1;

else

% Change to a 1x2 grid

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnWidth = {245, '1x'};

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

end

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.AutoResizeChildren = 'off';

app.UIFigure.Position = [100 100 650 516];

app.UIFigure.Name = 'MATLAB App';

app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

% Create GridLayout

app.GridLayout = uigridlayout(app.UIFigure);

app.GridLayout.ColumnWidth = {245, '1x'};

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnSpacing = 0;

app.GridLayout.RowSpacing = 0;

app.GridLayout.Padding = [0 0 0 0];

app.GridLayout.Scrollable = 'on';

% Create LeftPanel

app.LeftPanel = uipanel(app.GridLayout);

app.LeftPanel.Layout.Row = 1;

app.LeftPanel.Layout.Column = 1;

% Create SelectImageButton

app.SelectImageButton = uibutton(app.LeftPanel, 'push');

app.SelectImageButton.ButtonPushedFcn = createCallbackFcn(app, @SelectImageButtonPushed, true);

app.SelectImageButton.Position = [67 488 100 22];

app.SelectImageButton.Text = 'Select Image';

% Create KernelSpinnerLabel

app.KernelSpinnerLabel = uilabel(app.LeftPanel);

app.KernelSpinnerLabel.HorizontalAlignment = 'right';

app.KernelSpinnerLabel.Enable = 'off';

app.KernelSpinnerLabel.Position = [12 452 40 22];

app.KernelSpinnerLabel.Text = 'Kernel';

% Create KernelSpinner

app.KernelSpinner = uispinner(app.LeftPanel);

app.KernelSpinner.Step = 2;

app.KernelSpinner.Limits = [1 50];

app.KernelSpinner.RoundFractionalValues = 'on';

app.KernelSpinner.Enable = 'off';

app.KernelSpinner.Position = [67 452 100 22];

app.KernelSpinner.Value = 15;

% Create OmegaSpinnerLabel

app.OmegaSpinnerLabel = uilabel(app.LeftPanel);

app.OmegaSpinnerLabel.HorizontalAlignment = 'right';

app.OmegaSpinnerLabel.Enable = 'off';

app.OmegaSpinnerLabel.Position = [7 423 45 22];

app.OmegaSpinnerLabel.Text = 'Omega';

% Create OmegaSpinner

app.OmegaSpinner = uispinner(app.LeftPanel);

app.OmegaSpinner.Step = 2;

app.OmegaSpinner.Limits = [1 50];

app.OmegaSpinner.RoundFractionalValues = 'on';

app.OmegaSpinner.Enable = 'off';

app.OmegaSpinner.Position = [67 423 100 22];

app.OmegaSpinner.Value = 15;

% Create wEditFieldLabel

app.wEditFieldLabel = uilabel(app.LeftPanel);

app.wEditFieldLabel.HorizontalAlignment = 'right';

app.wEditFieldLabel.Enable = 'off';

app.wEditFieldLabel.Position = [14 282 25 22];

app.wEditFieldLabel.Text = 'w';

% Create wEditField

app.wEditField = uieditfield(app.LeftPanel, 'numeric');

app.wEditField.Limits = [0.5 1];

app.wEditField.Enable = 'off';

app.wEditField.Position = [54 282 39 22];

app.wEditField.Value = 0.95;

% Create pEditFieldLabel

app.pEditFieldLabel = uilabel(app.LeftPanel);

app.pEditFieldLabel.HorizontalAlignment = 'right';

app.pEditFieldLabel.Enable = 'off';

app.pEditFieldLabel.Position = [140 282 25 22];

app.pEditFieldLabel.Text = 'p';

% Create pEditField

app.pEditField = uieditfield(app.LeftPanel, 'numeric');

app.pEditField.Limits = [0.5 1];

app.pEditField.Enable = 'off';

app.pEditField.Position = [180 282 39 22];

app.pEditField.Value = 0.95;

% Create WSwitchLabel

app.WSwitchLabel = uilabel(app.LeftPanel);

app.WSwitchLabel.HorizontalAlignment = 'center';

app.WSwitchLabel.Enable = 'off';

app.WSwitchLabel.Position = [9 250 25 22];

app.WSwitchLabel.Text = 'W';

% Create WSwitch

app.WSwitch = uiswitch(app.LeftPanel, 'slider');

app.WSwitch.Orientation = 'vertical';

app.WSwitch.Enable = 'off';

app.WSwitch.Position = [12 179 20 45];

% Create VSwitchLabel

app.VSwitchLabel = uilabel(app.LeftPanel);

app.VSwitchLabel.HorizontalAlignment = 'center';

app.VSwitchLabel.Enable = 'off';

app.VSwitchLabel.Position = [46 250 25 22];

app.VSwitchLabel.Text = 'V';

% Create VSwitch

app.VSwitch = uiswitch(app.LeftPanel, 'slider');

app.VSwitch.Orientation = 'vertical';

app.VSwitch.Enable = 'off';

app.VSwitch.Position = [50 179 20 45];

% Create V\_RSwitchLabel

app.V\_RSwitchLabel = uilabel(app.LeftPanel);

app.V\_RSwitchLabel.HorizontalAlignment = 'center';

app.V\_RSwitchLabel.Enable = 'off';

app.V\_RSwitchLabel.Position = [118 251 29 22];

app.V\_RSwitchLabel.Text = 'V\_R';

% Create V\_RSwitch

app.V\_RSwitch = uiswitch(app.LeftPanel, 'slider');

app.V\_RSwitch.Orientation = 'vertical';

app.V\_RSwitch.Enable = 'off';

app.V\_RSwitch.Position = [123 179 20 45];

% Create RunButton

app.RunButton = uibutton(app.LeftPanel, 'push');

app.RunButton.ButtonPushedFcn = createCallbackFcn(app, @RunButtonPushed, true);

app.RunButton.Enable = 'off';

app.RunButton.Position = [103 46 127 85];

app.RunButton.Text = 'Run';

% Create Sigma\_tSliderLabel

app.Sigma\_tSliderLabel = uilabel(app.LeftPanel);

app.Sigma\_tSliderLabel.HorizontalAlignment = 'right';

app.Sigma\_tSliderLabel.Enable = 'off';

app.Sigma\_tSliderLabel.Position = [8 342 50 22];

app.Sigma\_tSliderLabel.Text = 'Sigma\_t';

% Create Sigma\_t

app.Sigma\_t = uislider(app.LeftPanel);

app.Sigma\_t.Limits = [0.01 100];

app.Sigma\_t.MajorTicks = [0.01 20 40 60 80 100];

app.Sigma\_t.Enable = 'off';

app.Sigma\_t.Position = [79 351 150 3];

app.Sigma\_t.Value = 20;

% Create Sigma\_rSlider\_2Label

app.Sigma\_rSlider\_2Label = uilabel(app.LeftPanel);

app.Sigma\_rSlider\_2Label.HorizontalAlignment = 'right';

app.Sigma\_rSlider\_2Label.Enable = 'off';

app.Sigma\_rSlider\_2Label.Position = [8 396 50 22];

app.Sigma\_rSlider\_2Label.Text = 'Sigma\_r';

% Create Sigma\_r

app.Sigma\_r = uislider(app.LeftPanel);

app.Sigma\_r.Limits = [0.01 100];

app.Sigma\_r.MajorTicks = [0.01 20 40 60 80 100];

app.Sigma\_r.Enable = 'off';

app.Sigma\_r.Position = [79 405 150 3];

app.Sigma\_r.Value = 20;

% Create RSwitchLabel

app.RSwitchLabel = uilabel(app.LeftPanel);

app.RSwitchLabel.HorizontalAlignment = 'center';

app.RSwitchLabel.Enable = 'off';

app.RSwitchLabel.Position = [84 251 25 22];

app.RSwitchLabel.Text = 'R';

% Create RSwitch

app.RSwitch = uiswitch(app.LeftPanel, 'slider');

app.RSwitch.Orientation = 'vertical';

app.RSwitch.Enable = 'off';

app.RSwitch.Position = [85 179 23 45];

% Create tSwitchLabel

app.tSwitchLabel = uilabel(app.LeftPanel);

app.tSwitchLabel.HorizontalAlignment = 'center';

app.tSwitchLabel.Enable = 'off';

app.tSwitchLabel.Position = [157 251 25 22];

app.tSwitchLabel.Text = 't';

% Create tSwitch

app.tSwitch = uiswitch(app.LeftPanel, 'slider');

app.tSwitch.Orientation = 'vertical';

app.tSwitch.Enable = 'off';

app.tSwitch.Position = [160 179 20 45];

% Create Adapt\_EQSwitchLabel

app.Adapt\_EQSwitchLabel = uilabel(app.LeftPanel);

app.Adapt\_EQSwitchLabel.HorizontalAlignment = 'center';

app.Adapt\_EQSwitchLabel.Enable = 'off';

app.Adapt\_EQSwitchLabel.Position = [38 77 61 22];

app.Adapt\_EQSwitchLabel.Text = 'Adapt\_EQ';

% Create Adapt\_EQSwitch

app.Adapt\_EQSwitch = uiswitch(app.LeftPanel, 'slider');

app.Adapt\_EQSwitch.Orientation = 'vertical';

app.Adapt\_EQSwitch.Enable = 'off';

app.Adapt\_EQSwitch.Position = [12 66 20 45];

% Create JSwitchLabel

app.JSwitchLabel = uilabel(app.LeftPanel);

app.JSwitchLabel.HorizontalAlignment = 'center';

app.JSwitchLabel.Enable = 'off';

app.JSwitchLabel.Position = [194 251 25 22];

app.JSwitchLabel.Text = 'J';

% Create JSwitch

app.JSwitch = uiswitch(app.LeftPanel, 'slider');

app.JSwitch.Orientation = 'vertical';

app.JSwitch.Enable = 'off';

app.JSwitch.Position = [197 179 20 45];

% Create RightPanel

app.RightPanel = uipanel(app.GridLayout);

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

% Create UIAxes\_2

app.UIAxes\_2 = uiaxes(app.RightPanel);

title(app.UIAxes\_2, 'Dehazed (J)')

app.UIAxes\_2.XColor = 'none';

app.UIAxes\_2.XTick = [];

app.UIAxes\_2.YColor = 'none';

app.UIAxes\_2.YTick = [];

app.UIAxes\_2.ZColor = 'none';

app.UIAxes\_2.GridColor = 'none';

app.UIAxes\_2.MinorGridColor = 'none';

app.UIAxes\_2.Position = [21 14 349 251];

% Create UIAxes

app.UIAxes = uiaxes(app.RightPanel);

title(app.UIAxes, 'Original')

app.UIAxes.XColor = 'none';

app.UIAxes.XTick = [];

app.UIAxes.YColor = 'none';

app.UIAxes.YTick = [];

app.UIAxes.ZColor = 'none';

app.UIAxes.GridColor = 'none';

app.UIAxes.MinorGridColor = 'none';

app.UIAxes.Position = [21 270 350 245];

% Show the figure after all components are created

app.UIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = FastDehazeImage\_exported

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

function outputImage = medianFilter(inputImage, windowSize)

% Get the size of the input image

[rows, cols] = size(inputImage);

% Calculate the padding size based on the window size

paddingSize = floor(windowSize / 2);

% Create a padded version of the input image

paddedImage = padarray(inputImage, [paddingSize, paddingSize]);

% Create an output image with the same size as the input image

outputImage = zeros(rows, cols);

% Apply the median filter

for i = 1:rows

for j = 1:cols

% Extract the window from the padded image

window = paddedImage(i:i+windowSize-1, j:j+windowSize-1);

% Calculate the median value of the window

medianValue = median(window(:));

% Set the output pixel value to the median value

outputImage(i, j) = medianValue;

end

end

%outputImage = uint8(outputImage);

end

function [output\_image ] = bilat\_filter(input\_image,radius,sigma\_s,sigma\_r )

% Compute the spatial weights

[X,Y]=meshgrid(-radius:radius,-radius:radius);

spatial\_weights = exp(-(X.^2+Y.^2)/(2\*sigma\_s^2));

% Create waitbar.

h = waitbar(0,'Wait,Applying bilateral filter...');

set(h,'Name','Bilateral Fiter Processing');

% Compute the size of the input and guidance images

[rows, cols] = size(input\_image);

% Initialize the output image

output\_image = zeros(size(input\_image));

% Apply the improved joint bilateral filter

for i = 1:rows

for j = 1:cols

% Compute the local region boundaries

row\_min = max(i - radius, 1);

row\_max = min(i + radius, rows);

col\_min = max(j - radius, 1);

col\_max = min(j + radius, cols);

% Extract the local region of the input

local\_input = input\_image(row\_min:row\_max, col\_min:col\_max);

% Compute the range weights

range\_weights = exp(-(local\_input-input\_image(i, j)).^2 / (2 \* sigma\_r^2));

%Taking the product of the range and domain filter.

weights = spatial\_weights((row\_min:row\_max)-i+radius+1,(col\_min:col\_max)-j+radius+1).\*range\_weights;

output\_image(i,j) = sum(weights(:).\*local\_input(:))/sum(weights(:));

end

waitbar(i/rows);

end

% Close waitbar.

close(h);

end

function atmosphericLight = estimateAtmosphericLight(darkChannel)

% Determine the number of pixels to consider (0.2% of total pixels)

numPixels = floor(0.002 \* numel(darkChannel));

% Reshape the channel into a column vector and sort it in descending order

darkChannelSorted = sort(darkChannel(:), 'descend');

% Select the top pixels with the highest intensities

topPixels = darkChannelSorted(1:numPixels);

% Retrieve the atmospheric light from the hazy image

atmosphericLight = double(max(topPixels));

end

function output\_image = bilat\_filter\_joint(input\_image, guidance\_image,radius, sigma\_s, sigma\_r, sigma\_t)

% Compute the spatial weights

[X,Y]=meshgrid(-radius:radius,-radius:radius);

spatial\_weights = exp(-(X.^2+Y.^2)/(2\*sigma\_s^2));

% Create waitbar.

h = waitbar(0,'Wait ,Applying joint bilateral filter...');

set(h,'Name','Joint Bilateral Fiter Processing');

% Compute the size of the input and guidance images

[rows, cols] = size(input\_image);

% Initialize the output image

output\_image = zeros(size(input\_image));

% Apply the improved joint bilateral filter

for i = 1:rows

for j = 1:cols

% Compute the local region boundaries

row\_min = max(i - radius, 1);

row\_max = min(i + radius, rows);

col\_min = max(j - radius, 1);

col\_max = min(j + radius, cols);

% Extract the local region of the input and guidance images

local\_input = input\_image(row\_min:row\_max, col\_min:col\_max);

local\_guidance = guidance\_image(row\_min:row\_max, col\_min:col\_max);

% Compute the range weights

range\_weights = exp(-(local\_input-input\_image(i, j)).^2 / (2 \* sigma\_r^2));

% Compute the temporal weights

temporal\_weights = exp(-(local\_guidance-guidance\_image(i, j)).^2 / (2 \* sigma\_t^2));

% Compute the combined weights

weights = spatial\_weights((row\_min:row\_max)-i+radius+1,(col\_min:col\_max)-j+radius+1).\*range\_weights.\*temporal\_weights;

% Apply the weighted average to filter the input image

output\_image(i, j) = sum(weights(:).\*local\_input(:)) / sum(weights(:));

end

waitbar(i/rows);

end

% Close waitbar.

close(h);

end

[*Published with MATLAB® R2023a*](https://www.mathworks.com/products/matlab/)

# Appendix B

classdef HomomorphicDehazeImage < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

GridLayout matlab.ui.container.GridLayout

LeftPanel matlab.ui.container.Panel

Image matlab.ui.control.Image

HistogramButton matlab.ui.control.Button

Dehazed\_RGBSwitchLabel matlab.ui.control.Label

Dehazed\_GraySwitch matlab.ui.control.Switch

Dehazed\_GraySwitchLabel matlab.ui.control.Label

GI\_FFTSwitch matlab.ui.control.Switch

GI\_FFTSwitchLabel matlab.ui.control.Label

TabGroup matlab.ui.container.TabGroup

GausHPTab matlab.ui.container.Tab

CEditField matlab.ui.control.NumericEditField

CEditFieldLabel matlab.ui.control.Label

gHEditField matlab.ui.control.NumericEditField

gHEditFieldLabel matlab.ui.control.Label

gLEditField matlab.ui.control.NumericEditField

gLEditFieldLabel matlab.ui.control.Label

D0EditField matlab.ui.control.NumericEditField

D0EditFieldLabel matlab.ui.control.Label

ButterHPTab matlab.ui.container.Tab

NEditField matlab.ui.control.NumericEditField

NEditFieldLabel matlab.ui.control.Label

D0EditField\_2 matlab.ui.control.NumericEditField

D0EditField\_2Label matlab.ui.control.Label

IdealHPTab matlab.ui.container.Tab

D0EditField\_3 matlab.ui.control.NumericEditField

D0EditField\_3Label matlab.ui.control.Label

Adapt\_EQSwitch matlab.ui.control.Switch

Adapt\_EQSwitchLabel matlab.ui.control.Label

RunButton matlab.ui.control.Button

I\_FFTSwitch matlab.ui.control.Switch

I\_FFTSwitchLabel matlab.ui.control.Label

DispSelectedFilterButton matlab.ui.control.Button

SelectImageButton matlab.ui.control.Button

Dehazed\_RGBSwitch matlab.ui.control.Switch

RightPanel matlab.ui.container.Panel

UIAxes matlab.ui.control.UIAxes

UIAxes\_2 matlab.ui.control.UIAxes

end

% Properties that correspond to apps with auto-reflow

properties (Access = private)

onePanelWidth = 576;

end

properties (Access = private)

imageData = []

I\_gray =[]

I\_gray\_defog = []

H = []

H\_selected =[]

I\_defog = []

imageJ\_adapt = []

end

methods (Access = private)

function Update\_H(app)

app.Dehazed\_GraySwitch.Enable='on';app.Dehazed\_GraySwitchLabel.Enable='on';

app.Dehazed\_RGBSwitch.Enable='on';app.Dehazed\_RGBSwitchLabel.Enable='on';

app.Adapt\_EQSwitch.Enable='on';app.Adapt\_EQSwitchLabel.Enable='on';

app.GI\_FFTSwitch.Enable='on';app.GI\_FFTSwitchLabel.Enable='on';

app.I\_FFTSwitch.Enable='on';app.I\_FFTSwitchLabel.Enable='on';

app.DispSelectedFilterButton.Enable='on';

app.RunButton.Enable='on';

end

end

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: SelectImageButton

function SelectImageButtonPushed(app, event)

close all;

[file ,path] = uigetfile({'\*.jpg;\*.jpeg;\*.png;\*.gif;\*.tif';'\*.\*'},'File Selector');

selectedfile = fullfile(path,file);

% Check if a file was selected

if isequal(file, 0) % No file selected

msgbox('There is NO image selected');

else

% Read the image file

app.imageData=imread(selectedfile);

imshow(app.imageData, 'Parent', app.UIAxes);

[height, width, ~] = size(app.imageData);

app.UIAxes.XLim = [0 width];

app.UIAxes.YLim = [0 height];

app.TabGroup.Visible='on';

app.RunButton.Enable='on';

app.HistogramButton.Enable='on';

app.imageData = im2double(app.imageData);

app.I\_gray = rgb2gray(app.imageData);

end

end

% Button pushed function: RunButton

function RunButtonPushed(app, event)

close all;

[app.I\_gray\_defog ,If,G\_I\_FFT] = homomorphic\_filter(app.I\_gray,app.H);

if strcmp(app.I\_FFTSwitch.Value,'On')

figure,imshow(If),title('I\_{FFT}');

end

if strcmp(app.GI\_FFTSwitch.Value,'On')

figure,imshow(G\_I\_FFT),title('G\*I\_{FFT}');

end

if strcmp(app.Dehazed\_GraySwitch.Value,'On')

figure,imshow(app.I\_gray\_defog),title('Dehazed Gray');

end

app.I\_defog = zeros(size(app.imageData));

for i = 1:3

app.I\_defog(:,:,i) = app.imageData(:,:,i).\*app.I\_gray\_defog;

end

app.I\_defog = rescale(app.I\_defog);

imshow(app.I\_defog, 'Parent', app.UIAxes\_2);

[height, width, ~] = size(app.I\_defog);

app.UIAxes\_2.XLim = [0 width];

app.UIAxes\_2.YLim = [0 height];

if strcmp(app.Dehazed\_RGBSwitch.Value,'On')

figure,imshow(app.I\_defog),title('Dehazed');

end

if strcmp(app.Adapt\_EQSwitch.Value,'On')

LAB = rgb2lab(uint8(app.I\_defog\*255));

L = LAB(:,:,1)/100;

L = adapthisteq(L,'NumTiles',[8 8],'ClipLimit',0.005);

LAB(:,:,1) = L\*100;

app.imageJ\_adapt = lab2rgb(LAB);

figure,imshow(app.imageJ\_adapt),title('J\_{adaptEQ}');

end

end

% Callback function: CEditField, D0EditField, GausHPTab, GausHPTab,

%

% ...and 2 other components

function GausHPTabButtonDown(app, event)

gL = app.gLEditField.Value;

gH = app.gHEditField.Value;

C = app.CEditField.Value;

D0 = app.D0EditField.Value;

app.H = gaushp(app.I\_gray, gL, gH, D0, C);

app.Update\_H()

end

% Callback function: ButterHPTab, ButterHPTab, D0EditField\_2,

% ...and 1 other component

function ButterHPTabButtonDown(app, event)

n = app.NEditField.Value;

D0 = app.D0EditField\_2.Value;

app.H = butterhp(app.I\_gray, D0,n);

app.Update\_H()

end

% Callback function: D0EditField\_3, IdealHPTab, IdealHPTab

function IdealHPTabButtonDown(app, event)

D0 = app.D0EditField\_3.Value;

app.H = idealhp(app.I\_gray, D0);

app.Update\_H()

end

% Button pushed function: DispSelectedFilterButton

function DispSelectedFilterButtonPushed(app, event)

close all;

plotTFSurface(app.H);

end

% Button pushed function: HistogramButton

function HistogramButtonPushed(app, event)

figure;imhist(app.I\_gray);title('Histogram');

end

% Changes arrangement of the app based on UIFigure width

function updateAppLayout(app, event)

currentFigureWidth = app.UIFigure.Position(3);

if(currentFigureWidth <= app.onePanelWidth)

% Change to a 2x1 grid

app.GridLayout.RowHeight = {516, 516};

app.GridLayout.ColumnWidth = {'1x'};

app.RightPanel.Layout.Row = 2;

app.RightPanel.Layout.Column = 1;

else

% Change to a 1x2 grid

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnWidth = {248, '1x'};

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

end

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Get the file path for locating images

pathToMLAPP = fileparts(mfilename('fullpath'));

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.AutoResizeChildren = 'off';

app.UIFigure.Position = [100 100 650 516];

app.UIFigure.Name = 'MATLAB App';

app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

% Create GridLayout

app.GridLayout = uigridlayout(app.UIFigure);

app.GridLayout.ColumnWidth = {248, '1x'};

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnSpacing = 0;

app.GridLayout.RowSpacing = 0;

app.GridLayout.Padding = [0 0 0 0];

app.GridLayout.Scrollable = 'on';

% Create LeftPanel

app.LeftPanel = uipanel(app.GridLayout);

app.LeftPanel.Layout.Row = 1;

app.LeftPanel.Layout.Column = 1;

% Create Dehazed\_RGBSwitch

app.Dehazed\_RGBSwitch = uiswitch(app.LeftPanel, 'slider');

app.Dehazed\_RGBSwitch.Enable = 'off';

app.Dehazed\_RGBSwitch.Position = [150 211 45 20];

% Create SelectImageButton

app.SelectImageButton = uibutton(app.LeftPanel, 'push');

app.SelectImageButton.ButtonPushedFcn = createCallbackFcn(app, @SelectImageButtonPushed, true);

app.SelectImageButton.Position = [74 483 100 23];

app.SelectImageButton.Text = 'Select Image';

% Create DispSelectedFilterButton

app.DispSelectedFilterButton = uibutton(app.LeftPanel, 'push');

app.DispSelectedFilterButton.ButtonPushedFcn = createCallbackFcn(app, @DispSelectedFilterButtonPushed, true);

app.DispSelectedFilterButton.Enable = 'off';

app.DispSelectedFilterButton.Position = [68 323 120 22];

app.DispSelectedFilterButton.Text = 'Disp Selected Filter';

% Create I\_FFTSwitchLabel

app.I\_FFTSwitchLabel = uilabel(app.LeftPanel);

app.I\_FFTSwitchLabel.HorizontalAlignment = 'center';

app.I\_FFTSwitchLabel.Enable = 'off';

app.I\_FFTSwitchLabel.Position = [43 293 37 22];

app.I\_FFTSwitchLabel.Text = 'I\_FFT';

% Create I\_FFTSwitch

app.I\_FFTSwitch = uiswitch(app.LeftPanel, 'slider');

app.I\_FFTSwitch.Enable = 'off';

app.I\_FFTSwitch.Position = [42 267 45 20];

% Create RunButton

app.RunButton = uibutton(app.LeftPanel, 'push');

app.RunButton.ButtonPushedFcn = createCallbackFcn(app, @RunButtonPushed, true);

app.RunButton.Enable = 'off';

app.RunButton.Position = [114 120 110 65];

app.RunButton.Text = 'Run';

% Create Adapt\_EQSwitchLabel

app.Adapt\_EQSwitchLabel = uilabel(app.LeftPanel);

app.Adapt\_EQSwitchLabel.HorizontalAlignment = 'center';

app.Adapt\_EQSwitchLabel.Enable = 'off';

app.Adapt\_EQSwitchLabel.Position = [38 144 61 22];

app.Adapt\_EQSwitchLabel.Text = 'Adapt\_EQ';

% Create Adapt\_EQSwitch

app.Adapt\_EQSwitch = uiswitch(app.LeftPanel, 'slider');

app.Adapt\_EQSwitch.Orientation = 'vertical';

app.Adapt\_EQSwitch.Enable = 'off';

app.Adapt\_EQSwitch.Position = [12 133 20 45];

% Create TabGroup

app.TabGroup = uitabgroup(app.LeftPanel);

app.TabGroup.Visible = 'off';

app.TabGroup.Position = [12 352 218 94];

% Create GausHPTab

app.GausHPTab = uitab(app.TabGroup);

app.GausHPTab.SizeChangedFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

app.GausHPTab.Title = 'GausHP';

app.GausHPTab.ButtonDownFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

% Create D0EditFieldLabel

app.D0EditFieldLabel = uilabel(app.GausHPTab);

app.D0EditFieldLabel.HorizontalAlignment = 'right';

app.D0EditFieldLabel.Position = [12 38 25 22];

app.D0EditFieldLabel.Text = 'D0';

% Create D0EditField

app.D0EditField = uieditfield(app.GausHPTab, 'numeric');

app.D0EditField.Limits = [1 100];

app.D0EditField.ValueChangedFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

app.D0EditField.Position = [52 38 30 22];

app.D0EditField.Value = 1;

% Create gLEditFieldLabel

app.gLEditFieldLabel = uilabel(app.GausHPTab);

app.gLEditFieldLabel.HorizontalAlignment = 'right';

app.gLEditFieldLabel.Position = [130 39 25 22];

app.gLEditFieldLabel.Text = 'gL';

% Create gLEditField

app.gLEditField = uieditfield(app.GausHPTab, 'numeric');

app.gLEditField.Limits = [0.09 1];

app.gLEditField.ValueChangedFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

app.gLEditField.Position = [170 39 30 22];

app.gLEditField.Value = 0.1;

% Create gHEditFieldLabel

app.gHEditFieldLabel = uilabel(app.GausHPTab);

app.gHEditFieldLabel.HorizontalAlignment = 'right';

app.gHEditFieldLabel.Position = [130 6 25 22];

app.gHEditFieldLabel.Text = 'gH';

% Create gHEditField

app.gHEditField = uieditfield(app.GausHPTab, 'numeric');

app.gHEditField.Limits = [1 4];

app.gHEditField.ValueChangedFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

app.gHEditField.Position = [170 6 30 22];

app.gHEditField.Value = 1.1;

% Create CEditFieldLabel

app.CEditFieldLabel = uilabel(app.GausHPTab);

app.CEditFieldLabel.HorizontalAlignment = 'right';

app.CEditFieldLabel.Position = [12 6 25 22];

app.CEditFieldLabel.Text = 'C';

% Create CEditField

app.CEditField = uieditfield(app.GausHPTab, 'numeric');

app.CEditField.Limits = [1 100];

app.CEditField.ValueChangedFcn = createCallbackFcn(app, @GausHPTabButtonDown, true);

app.CEditField.Position = [52 6 30 22];

app.CEditField.Value = 2;

% Create ButterHPTab

app.ButterHPTab = uitab(app.TabGroup);

app.ButterHPTab.SizeChangedFcn = createCallbackFcn(app, @ButterHPTabButtonDown, true);

app.ButterHPTab.Title = 'ButterHP';

app.ButterHPTab.ButtonDownFcn = createCallbackFcn(app, @ButterHPTabButtonDown, true);

% Create D0EditField\_2Label

app.D0EditField\_2Label = uilabel(app.ButterHPTab);

app.D0EditField\_2Label.HorizontalAlignment = 'right';

app.D0EditField\_2Label.Position = [74 41 25 22];

app.D0EditField\_2Label.Text = 'D0';

% Create D0EditField\_2

app.D0EditField\_2 = uieditfield(app.ButterHPTab, 'numeric');

app.D0EditField\_2.Limits = [1 100];

app.D0EditField\_2.ValueChangedFcn = createCallbackFcn(app, @ButterHPTabButtonDown, true);

app.D0EditField\_2.Position = [114 41 30 22];

app.D0EditField\_2.Value = 1;

% Create NEditFieldLabel

app.NEditFieldLabel = uilabel(app.ButterHPTab);

app.NEditFieldLabel.HorizontalAlignment = 'right';

app.NEditFieldLabel.Position = [74 11 25 22];

app.NEditFieldLabel.Text = 'N';

% Create NEditField

app.NEditField = uieditfield(app.ButterHPTab, 'numeric');

app.NEditField.Limits = [1 100];

app.NEditField.RoundFractionalValues = 'on';

app.NEditField.ValueChangedFcn = createCallbackFcn(app, @ButterHPTabButtonDown, true);

app.NEditField.Position = [114 11 30 22];

app.NEditField.Value = 2;

% Create IdealHPTab

app.IdealHPTab = uitab(app.TabGroup);

app.IdealHPTab.SizeChangedFcn = createCallbackFcn(app, @IdealHPTabButtonDown, true);

app.IdealHPTab.Title = 'IdealHP';

app.IdealHPTab.ButtonDownFcn = createCallbackFcn(app, @IdealHPTabButtonDown, true);

% Create D0EditField\_3Label

app.D0EditField\_3Label = uilabel(app.IdealHPTab);

app.D0EditField\_3Label.HorizontalAlignment = 'right';

app.D0EditField\_3Label.Position = [70 25 25 22];

app.D0EditField\_3Label.Text = 'D0';

% Create D0EditField\_3

app.D0EditField\_3 = uieditfield(app.IdealHPTab, 'numeric');

app.D0EditField\_3.Limits = [1 100];

app.D0EditField\_3.RoundFractionalValues = 'on';

app.D0EditField\_3.ValueChangedFcn = createCallbackFcn(app, @IdealHPTabButtonDown, true);

app.D0EditField\_3.Position = [110 25 30 22];

app.D0EditField\_3.Value = 1;

% Create GI\_FFTSwitchLabel

app.GI\_FFTSwitchLabel = uilabel(app.LeftPanel);

app.GI\_FFTSwitchLabel.HorizontalAlignment = 'center';

app.GI\_FFTSwitchLabel.Enable = 'off';

app.GI\_FFTSwitchLabel.Position = [144 292 52 22];

app.GI\_FFTSwitchLabel.Text = 'G\*I\_FFT';

% Create GI\_FFTSwitch

app.GI\_FFTSwitch = uiswitch(app.LeftPanel, 'slider');

app.GI\_FFTSwitch.Enable = 'off';

app.GI\_FFTSwitch.Position = [150 266 45 20];

% Create Dehazed\_GraySwitchLabel

app.Dehazed\_GraySwitchLabel = uilabel(app.LeftPanel);

app.Dehazed\_GraySwitchLabel.HorizontalAlignment = 'center';

app.Dehazed\_GraySwitchLabel.Enable = 'off';

app.Dehazed\_GraySwitchLabel.Position = [19 235 86 22];

app.Dehazed\_GraySwitchLabel.Text = 'Dehazed\_Gray';

% Create Dehazed\_GraySwitch

app.Dehazed\_GraySwitch = uiswitch(app.LeftPanel, 'slider');

app.Dehazed\_GraySwitch.Enable = 'off';

app.Dehazed\_GraySwitch.Position = [42 211 45 20];

% Create Dehazed\_RGBSwitchLabel

app.Dehazed\_RGBSwitchLabel = uilabel(app.LeftPanel);

app.Dehazed\_RGBSwitchLabel.HorizontalAlignment = 'center';

app.Dehazed\_RGBSwitchLabel.Enable = 'off';

app.Dehazed\_RGBSwitchLabel.Position = [127 234 86 22];

app.Dehazed\_RGBSwitchLabel.Text = 'Dehazed\_RGB';

% Create HistogramButton

app.HistogramButton = uibutton(app.LeftPanel, 'push');

app.HistogramButton.ButtonPushedFcn = createCallbackFcn(app, @HistogramButtonPushed, true);

app.HistogramButton.Enable = 'off';

app.HistogramButton.Position = [74 453 100 22];

app.HistogramButton.Text = 'Histogram';

% Create Image

app.Image = uiimage(app.LeftPanel);

app.Image.Position = [24 7 199 106];

app.Image.ImageSource = fullfile(pathToMLAPP, '2023-05-26\_18h16\_08.png');

% Create RightPanel

app.RightPanel = uipanel(app.GridLayout);

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

% Create UIAxes\_2

app.UIAxes\_2 = uiaxes(app.RightPanel);

title(app.UIAxes\_2, 'Dehazed')

app.UIAxes\_2.XColor = 'none';

app.UIAxes\_2.XTick = [];

app.UIAxes\_2.YColor = 'none';

app.UIAxes\_2.YTick = [];

app.UIAxes\_2.ZColor = 'none';

app.UIAxes\_2.GridColor = 'none';

app.UIAxes\_2.MinorGridColor = 'none';

app.UIAxes\_2.Position = [25 12 350 250];

% Create UIAxes

app.UIAxes = uiaxes(app.RightPanel);

title(app.UIAxes, 'Original')

app.UIAxes.XColor = 'none';

app.UIAxes.XTick = [];

app.UIAxes.YColor = 'none';

app.UIAxes.YTick = [];

app.UIAxes.ZColor = 'none';

app.UIAxes.GridColor = 'none';

app.UIAxes.MinorGridColor = 'none';

app.UIAxes.Position = [25 265 350 250];

% Show the figure after all components are created

app.UIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = HomomorphicDehazeImage

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

function [Iout ,If,G] = homomorphic\_filter(I,H)

tic

I\_log = log(1 + I);

If = fft2(I\_log);

If = fftshift(If);

G = H.\*If;

Iout = real(ifft2(ifftshift(G)));

Iout = exp(Iout) - 1;

toc

end

function H = gaushp(I, gL, gH, D0, C)

%gL - low values

%gH - high values

%D0 - size of the gaussian, affects more/less frequencies

%C - as with D0, it affects the exponential in H. It is like a tug of war

% between D0 and C, working in different ranges. D0 is a sharpness factor

% for the big changes in the image whereas c is sharpening in a small range.

[M,N] = size(I);

% Generate a meshgrid for the frequency domain

[X, Y] = meshgrid(1:N,1:M);

gaussianNumerator = (X - floor(N/2)).^2 + (Y - floor(M/2)).^2;

H = (gH-gL)\*(1-exp(-C\*gaussianNumerator./(D0.^2)))+gL;

end

function [out]=butterhp(I, D0,n)

%D0 - BHPF passes all the frequencies greater than D\_{0} value without attenuation and cuts off all the frequencies less than it. The transition point between H(u, v) = 1 and H(u, v) = 0, so this is termed as cutoff frequency. But instead of making a sharp cut-off(like, Ideal High pass Filter (IHPF)), it introduces a smooth transition from 0 to 1 to reduce ringing artifacts.

%n - filter order value

[M,N] = size(I);

% Generate a meshgrid for the frequency domain

[X, Y] = meshgrid(1:N,1:M);

D = sqrt((X - floor(N/2)).^2 + (Y - floor(M/2)).^2);

out=1./(1.+(D0./D).^(2\*n));

end

function H = idealhp(I, cutoff)

% size: size of the filter in pixels

% cutoff: frequency cutoff for the high-pass filter

[M,N] = size(I);

% Generate a meshgrid for the frequency domain

[X, Y] = meshgrid(1:N,1:M);

D = ((X - floor(N/2)).^2 + (Y - floor(M/2)).^2);

% Create the ideal high-pass filter

H = double(D >= cutoff);

end

function plotTFSurface(Z)

[M,N] = size(Z);

figure,imshow(Z,[]),title('H - TF');

% Zoom in on the center of the plot

ylim([floor(M/2)-100, floor(M/2)+100]);

xlim([floor(N/2)-100, floor(N/2)+100]);

% Create the surface plot

figure;

surf(Z,'FaceAlpha',0.5);

xlabel('X');

ylabel('Y');

zlabel('Transfer Function');

title('Surface Plot of Transfer Function');

% Zoom in on the center of the plot

ylim([floor(M/2)-100, floor(M/2)+100]);

xlim([floor(N/2)-100, floor(N/2)+100]);

end