

**פרויקט חילוץ דופק באמצעות מצלמת RGB ושימוש בטכניקת (Eulerian Video Magnification (EVM**

**מבצעים**:

יונתן עמיר, הנדסת מחשבים, ת.ז. 301314860, מייל: yonatanamir1@gmail.com

דורון ארמון, הנדסת מחשבים, ת.ז. 305745887, מייל: doronarmon@gmail.com

נריה מזוז, מדעי המחשב, ת.ז. 313347015, מייל: Neriya92@gmail.com

**מנחים**:

גיל שמאי ורון סלוסברג

ירון חונן - מהנדס המעבדה

**מועד**:

הפרויקט התבצע בסמסטר אביב 2016 וחורף 2017

1. **רקע - הטכנולוגיה הקיימת ובעיותיה**

* **ה"בעיה"**

קלינאים ומדענים מעוניינים בפיתוח שיטות אלטרנטיביות לחילוץ מדדים רפואיים בדרכים חדשות, כאשר בעדיפות עליונה נמצאים מדדים רפואיים חיוניים ("Vital Signs") הכוללים דופק, קצב ואופי נשימתי, סטורציה, טמפרטורה ולחץ דם. המטרה של הפרויקט היא להציע אלטרנטיבה לשיטת המדידה הקונבנציונלית הנעשית בבתי חולים של מדידת דופק של פציינט באמצעות Pulse Oximeter פיזי המחובר בשלב הטריאז' לאצבע המורה של הנבדק.

שיטה זאת יכולה להחליף הן את מדידת הדופק במהלך שלבי הקבלה של הפציינט למוסד הרפואי והן עבור monitoring מתמשך אחרי אשפוז הפציינט באחת ממחלקות בית החולים או "מיון שוכבים".

רוב הבדיקות והמדידות הרפואיות המרכזיות כוללות קיום מגע כלשהו עם הפציינט בין אם במגע לא פולשני (contacts based non-invasive) או בצורה פולשנית (invasive). תוצר הלוואי של השיטות הקיימות למדידת מדדים חיוניים ("Vital Signs") מבוססי מגע הוא קילוף או תלישה של עור בכל פעם ש- patches רפואיים מוסרים מהמטופל ובכלליות גרימת חוסר נוחות כאשר המדדים החיוניים צריכים להיאסף ב- monitoring לאורך ימים או שבועות באופן רציף. שימוש בשיטה של חילוץ דופק ללא מגע יכולה לייתר את הציוד ה"חומרתי" הקיים בבתי החולים ע"י שימוש במצלמת RGB פשוטה אשר עלותה נמוכה. שיטה זאת מציעה כמה יתרונות מרכזיים: מדידת דופק מוגבלת לרוב פיזיולוגית רק לאצבע המורה של הפציינט בכף היד באופן בו המכשיר הרפואי יכול ליפול כתוצאה מתזוזת החולה ו/או להינזק במהלך monitoring שלוקח זמן רב בצורה כזו אשר תפגע בתוצאות המדידה החיונית. ניתן לעקוף את חסרונות אלו ע"י שימוש בטכניקות אנליזה של הסיגנלים של הוידאו כפי שיודגמו בהמשך, אשר לא מסתמכות על Pulse Oximeter סטנדרטי למדידת הדופק של המטופל אלא במצלמת RGB סטנדרטית של מחשב נייד בעל frame rate הקיים ברוב המצלמות הזמינות למשתמשים שונים (טלפונים, לפטופים וכו'). ע"י שימוש במצלמה ולא בהתקן חומרתי ניתן למדוד דופק מאזורים שונים בגוף בצורה יעילה ומהירה, כפי שיודגם בהמשך חילוץ הדופק שלנו מאזור הפנים של המטופל ללא מגע. הגישה בה בחרנו לצורך חילוץ הדופק נקראת Eulerian Video Magnification או בקיצור EVM אשר פורסמה לראשונה ע"י צוות מפתחים מאוניברסיטת MIT אשר בסיסה הוא העצמת תנועות וצבע מ- input של קטעי וידאו סטנדרטיים.

**2. הפתרון שלנו – כיצד מתגבר על בעיות הקימות, והחדשנות שבו**

* **מטרת/ הגדרת הפרויקט:** 
  + **חלק א' - לימוד והבנת רקע תיאורטי של המאמר של חוקרי MIT בנושא שיטת EVM.**
  + **חלק ב'-**
    - **בהתבסס על אותו המאמר - יישום השיטה עבור חילוץ דופק בזמן אמת עם מצלמת RGB (מידע אשר לא מפורט במאמר).**
    - **יצירת 3 פונקציות נוספות בתחום עיבוד תמונה עבור "פציינט בחדר מיון"**
* שלב א' (עד מצגת אמצע)
  + קריאה ולימוד תיאורטי של יסודות עיבוד תמונה
  + הבנה תיאורטית של המאמר הספציפי בנושא EVM
  + יישום חילוץ דופק (offline) באמצעות MATLAB
  + הצגת התוצאות המתקבלות תוך השוואה לתוצאות הצוות המפרסם מ-MIT על סרטונים זהים
  + ויזואליזציה של video magnification מאזור הפנים (וידאו)
* שלב ב'
  + יצירת הפרויקט (של חילוץ הדופק) מחדש ב- ++C (זמן אמת)
  + יצירת אלגוריתם ראשון - חילוץ דופק המתבסס על שיטת EVM מהמאמר + הדוקטורט של רובינשטיין מ-MIT (בשיטת Peak Detection)
  + שימוש במצלמה חיצונית עם קצב פריימים גבוהה (90fps) לשיפור דיוק
  + יצירת אלגוריתם שני - חילוץ דופק המתבסס על שיטת EVM והתמרות פוריה (DFT) למציאת התדר המתאים לדופק הקיים של ה"פציינט"
  + יצירת פאטצ' מתאים (בחירת אזור עניין ממוקד יותר בתוך חלקים בפנים)
  + שימוש ב- PCA לצורך שיפור רמת הדיוק
  + לצורך בניית ממשק גרפי לפרויקט - יצירת threads לשיפור ביצועים: הראשון עבור חישוב דופק והשני עבור הצגה ויזואלית בממשק ב- real-time.
  + שימוש ב- face-recognition
  + שימוש ב- face-detaction (וידאו real-time)
  + שימוש ב-face-detaction (תמונות סטילס, "פספורט מתיק פציינט")
  + הזנת משתמשים חדשים בזמן אמת (face-recognition) ושמירת פרטיהם לצורך זיהוים מחדש באופן אוטומטי בעתיד
  + ניתוח הבעות פנים בזמן אמת: עיניים, פה
  + שערוך נתוני "פציינט" בזמן אמת - גיל, מין
  + מיפוי הפנים בזמן אמת (שימוש ב- 70 נקודות ציון)
  + יצירת תיק פציינט המכיל מידע רפואי היסטורי + חיבור למשתמש (face recognition)
  + יצירת פונקצית "הדמיית בדיקת רפואית" ב-Matlab עבור הרובוט
  + יצירת סרטון המתאר את הבעיה "טעויות רפואיות ודיאגנוסטיות"
  + יצירת סרטון המתאר את הבעיה "עומסים בחדרי מיון"
* **שימוש במאמרים עבור כלים יישומיים - עיבוד תמונה:**
  + המאמר הראשי שניתחנו (עד פרזנטציית אמצע) - MIT חילוץ EVM
  + מאמרים ומחקרים נוספים אותם עליהם התבסס היישום (קראנו והצלבנו מידע) כפי שמופיע בטבלה:

|  |  |  |  |
| --- | --- | --- | --- |
| **מס** | **שם מאמר** | **מה יישמנו מתוך המאמר** | **קישור** |
| 1 | מאמר ראשי (הדרישה עבור הפרויקט ליישום) - **EVM של MIT**  Eulerian Video Magnification for Revealing Subtle Changes in the World | * Temporal processing * Spatial decomposition * magnification - לצורך ויזואליזציה * ניתוח המאמר והבנה תיאורטית של EVM | <http://people.csail.mit.edu/mrub/papers/vidmag.pdf> |
| 2 | דוקטורט של רובינשטיין MIT  Analysis and Visualization of Temporal Variations in Video | * השיטה של רובינשטיין מעמוד 62 בתזה למציאת Peak (מדידת ה- peaks של כל פיקסל בנפרד+ממוצע על כל ה- peaks של הפיקסלים השונים לצורך קבלת הדופק בסופו של דבר) | <http://people.csail.mit.edu/mrub/papers/RubinsteinPhDThesis.pdf> |
| 3 | מאמר של צוות מגרמניה  Heart rate monitoring in ultra-high-eld MRI using frequency information obtained from video signals of the human skin compared to electrocardiography and pulse oximetry | * נסיון החלפת שיטת ה- peak detection עליה המליצו רובינשטיין וה"איראנים" ושימוש בהתמרות פוריה (DFT) במקום | <https://www.degruyter.com/view/j/cdbme.2015.1.issue-1/cdbme-2015-0018/cdbme-2015-0018.pdf> |
| 4 | מאמר של צוות מאיראן | * עפ"י המאמר, הומלץ להשתמש ב- Peak Detection | <https://www.researchgate.net/publication/269999734_RETRACTED_A_temporal_video-processing_method_to_improve_heart_rate_estimation> |
| 5 | מאמר של Xiaobai Li  Remote Heart Rate Measurement From Face Videos Under Realistic Situations | * שימוש ב- patch מאזור הסנטר * שימוש בערוץ הירוק * שימוש ב- PCA | <http://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Li_Remote_Heart_Rate_2014_CVPR_paper.pdf> |
| 6 | מאמר של Matt Estrada  Amplification of Heart Rate in Multi-Subject Videos | * שימוש ב- patch מאזור הסנטר * שימוש בערוץ הירוק | <https://web.stanford.edu/class/ee368/Project_Spring_1415/Reports/Stowers_Estrada.pdf> |
| 7 | שני דוקטורנטים מ- MIT עליהם רובינשטיין המליץ להסתמך בנוגע ל שיטת חילוץ הדופק  Eulerian Video Processing and Medical Applications | * נסיון לקיחת פאטצ'ים מאזור המצח * נסיון לקיחת פאטצ'ים מאזור הלחיים | <https://dspace.mit.edu/bitstream/handle/1721.1/77452/826647817-MIT.pdf?sequence=2> |

* **התייעצות על פיתוח:**
  + צוות מגרמניה (מהפיתוח שמוצג במאמר מס' 3)
  + אסף גלזר מנכ"ל nanit (התייעצות בנוגע לחשיבות fps ורזולוציה, פתרון בעיות נוספות)

**3. שיקולי תכנון, תכנונים שנעשו על ידי הקבוצה**

**אלגוריתם ראשון – לפי התזה של רובינשטיין (גרסת Offline):**

1. נבצע face detection על הפריים הראשון, ונחתוך את הוידאו לפי הקואורדינטות שהתקבלו.

**שלבים מתוך ה- EVM:**

2. נבצע פירמידת גאוס על כל הפריימים ונשמור את השכבה הגבוהה ביותר בפירמידה של פריים.

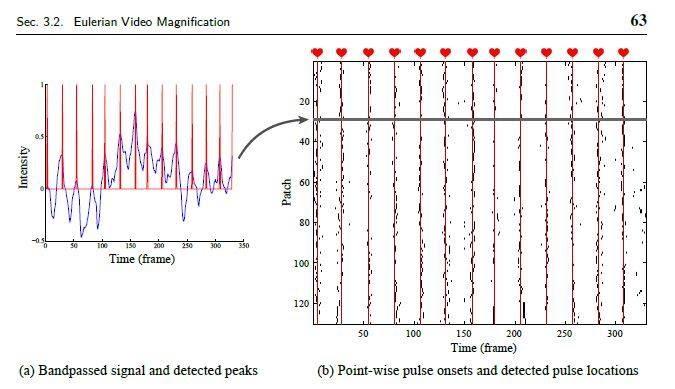
3. נבצע temporal filtering לתדרים בטווח מסוים שנקבע מראש.

**שלבים בהשראת התזה:**

4. יצירת חלונות בגודל של 15 פריימים עבור כל פיקסל. ערך הפיקסל יהיה עצמת ה- red channel (לבסוף חישבנו PCA ש"מתחשב" בשלושת ערוצי ה- RGB) שלו.

5. עבור כל חלון של כל פיקסל, מחפשים פיק, ושומרים את מס' הפריים שבו התרחש הפיק.

תמונה הממחישה זאת (מתוך התזה של רובינשטיין):



6. חישוב ממוצע קטום (חישוב ממוצע רק על השליש האמצעי של הערכים) על מיקומי הפיקים של כל פיקסל בחלון זמן מסוים, ובאמצעות כך גילוי באיזה פריים היה פיק עבור אותו חלון זמן.

7. חישוב של ממוצע הפרשי הזמן בין זוג פיקים עוקבים, ולאחר מכן חילוץ הדופק ע"י הנוסחה: .

**אלגוריתם שני – בהנחיית המנחים ובאמצעות מאמר שמצאנו[[1]](#footnote-1) (גרסת Offline):**

1. נבצע face detection על הפריים הראשון, ונחתוך את הוידאו לפי הקואורדינטות שהתקבלו.

**שלבים מתוך הvideo magnification:**

2. נבצע פירמידת גאוס על כל הפריימים ונשמור את השכבה הגבוהה ביותר בפירמידה של פריים

3. נבצע temporal filtering לתדרים בטווח 0.83-2.5 (שזה מתקשר לדופק בין 50 ל-150)

**שלבים בהשראת המאמר:**

4. ניצור חלונות זמן בגודל 5 שניות (5\*fps פריימים) עבור כל פיקסל. ערך הפיקסל יהיה ה- green channel שלו.

5. עבור כל חלון של ערכי פיקסל בחלון זמן מסוים נבצע Hamming window ונכפיל אותו עם החלון המקורי (איבר-איבר).

6. נחשב DFT על החלון שהתקבל.

7. מתוצאת ה- DFT נחלץ את התדר המקסימלי ונמיר אותו לדופק (BPM = 60\*Hz)

8. נעשה ממוצע קטום על כל ערכי הדופק של הפיקסלים שהתקבלו עבור חלון מסוים, והתוצאה שהתקבלה היא הדופק עבור חלון זמן זה.

**4. ניסויים - מהלך, שיטה, בעיות ואופן התגברות עליהן**

**שלב א'** - עד מצגת אמצע

* קריאה ולימוד תיאורטי של יסודות עיבוד התמונה
  + בשלב הראשון היה עלינו לקרוא ולנתח מאמר אקדמי ברמה גבוהה (פעם ראשונה בתואר)
  + תרגמנו את המאמר
  + התחלנו לרכז את כל המושגים התיאורטיים שלא הכרנו מתחום עיבוד התמונה ולקרוא עליהם בהרחבה במאמרים נוספים/אינטרנט/סרטונים וכו'
  + ניתחנו את המודלים המתמטיים וניסינו להבין מה עומד מאחוריהם (בהתבסס על ידע קודם שהיה לנו מקורסים שונים: מד"ח וטורי פוריה, אנליזה נומרית, אלגברה ואלגברה מודרנית וכו')
* הבנה תיאורטית של המאמר הספציפי בנושא EVM
  + הבנת העקרונות מאחורי השיטה להעצמת שינויים מזעריים - EVM
  + סקירת היישומים השונים הקיימים בעולם תוך שימוש באלגוריתם זה (שחזור שיחה ע"י ויברציות של עצם דומם, חילוץ דופק, חילוץ קצב נשימות, ייצוב תמונה הנובע מתזוזת מצלמת SLR וכו')
  + התייעצות ב- skype עם ד"ר אסף גלזר (מייסד nanit) בנוגע למאמרים השונים
  + קריאת הדוקטורט המלא של רובינשטיין על מנת להבין את תהליך יישום חילוץ הדופק שלא מופיע במאמר הראשי של MIT
  + השוואת השיטה האוילרית (ה- E שב- EVM) לשיטה הלגראנז'ית (אשר מוזכרות במאמר)
  + לימוד ושימוש בשיטת ה- Spatial Decomposition
  + לימוד ושימוש בשיטת ה- Temporal Processing
  + לימוד ושימוש בשיטת ה- Magnification and Reconstruction
  + **מושגים אותם למדנו לצורך הבנת המאמר (חלק מתוך קובץ נפרד בו תרגמנו את המאמר) - הסבר וקישור להרחבות עליהם**
    - spatial decomposition = פירוק מרחבי – לחלק את המרחב לתאים פשוטים <http://www.cs.duke.edu/courses/cps124/spring04/notes/12_datastructures/notes.pdf>
* temporal filtering = בניגוד לתמונה, בה סינון מרחבי יכול למשל לטשטש את התמונה או לחדד – סינון זמני עוסק בסינון של רצף תמונות (זאת אומרת, כביכול, שהאות המסונן הוא רצף התמונות והסינון הוא של כל פיקסל ופיקסל בתמונה לפי השתנותו בזמן).הסינון עצמו יכול לטשטש את השינויים הנעשים בחלוף הזמן של רצף התמונות (כמו תנועת עלה שזז ברוח למשל).<http://stackoverflow.com/questions/20091946/temporal-filtering-and-their-concepts-in-image-signal-video-processing>
* spatio-temporal sensitivity – רגישות לזיהוי תנועות לאורך זמן ברצף תמונות (בהקשר של המאמר, הכוונה היא שהעין האנושית מתאפיינת ביכולת נמוכה - רגישות נמוכה לתפיסת שינויים בתנועה בתדרים גבוהים).<http://delivery.acm.org/10.1145/390000/383748/p39-yee.pdf?ip=132.68.62.8&id=383748&acc=ACTIVE%20SERVICE&key=0D17F1A88EABC760.73E0BAC401C36D1F.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=626103589&CFTOKEN=73311259&__acm__=1465215853_1d7ccf66583e14da57d2d7f6f5501edc>
* time series – רצף של תצפיות שנאמדו זו אחר זו, באינטרוול זמן נתון.<https://he.wikipedia.org/wiki/%D7%A1%D7%93%D7%A8%D7%94_%D7%A2%D7%AA%D7%99%D7%AA>
* spatial pooling – דרך לחשב ייצוג לתמונה, כאשר כל פיקסל מחושב כפונקציה של הפיקסלים בסביבתו.<https://www.quora.com/What-is-spatial-pooling-in-computer-vision>
* quantization noise – "רעש" בתמונה שנובע מטעויות דגימה.<https://en.wikipedia.org/wiki/Quantization_(signal_processing)#Quantization_noise_model>
* brightness constancy assumption – זו ההנחה כי הבהירות באזור מסוים קטן מספיק, תישאר זהה לכל אורך רצף התמונות, למרות שייתכן ומיקום האזור ישתנה.
* optical flow – זהו זיהוי תנועה של אובייקט על פי בהירות ברצף של תמונות.  
  <http://cs.haifa.ac.il/hagit/courses/seminars/visionTopics/Presentations/Lecture06_Optical%20Flow.ppt>

<http://www.sci.utah.edu/~gerig/CS6320-S2013/Materials/CS6320-CV-S2012-OpticalFlow-I.pdf>

* preceptioaly appealing motion aggregation – מבחינה תפיסתית הגזמת תנועה מושכת / מעניינת (לא הסבר אלא תרגום ישיר מגוגל טרנסלייט).
* motion segmentation - משמש לאיתור אובייקטים שנעים, משנים את מיקומיהם, בתוך רצף של מס' תמונות. המשמעות היא הפרדת אובייקטים אלו מהרקע הסטטי.

<http://computervision.wikia.com/wiki/Motion_segmentation>

* image in-painting – שיטה ל"שחזור" חלקים חסרים או פגומים בתמונה, ובנוסף שיטה שבאמצעותה ניתן להוריד או לשנות אובייקטים לא רצויים בתמונה. השיטה מבצעת אינטרפולציה של הערכים שליד האובייקט הלא רצוי ומחליפה אותו בערכים אלה.

<https://en.wikipedia.org/wiki/Inpainting>

<http://www.slideshare.net/PulkitGoyal1/image-inpainting>

* Single number בניגוד ל bandpass filtering – כנראה (בקונטקסט של המאמר) ההבדל בין חילוץ הדופק כפרמטר מספרי יחיד לעומת חילוץ דופק וויזואליזציה שניתן לעשות בכלי החדש שפותח ומוצג במאמר.
* temporal aliasing - התחזות של תדרים אל תדרים אחרים בעקבות דגימה שלא עומדת בתדר נייקוויסט. ניתן לראות זאת גם באות אודיו וגם ברצף תמונות (וידאו) .

<https://en.wikipedia.org/wiki/Aliasing>

* non-photorealistic - קשור לתחום של אמנות דיגיטלית - עיבוד תמונות (כמו יצירת אפקטים) בצורה הדומה לשיטות שלא מנסות להציג את המציאות האמיתית כפי שהיא (כמו ציור, רישום, אנימציה).
* Low pass filtering - (ביטול תדרים מתחת לסף מקסימלי מסויים - התדרים הנלקחים הם רק התדרים שמעל הסף) - A low-pass filter is a [filter](https://en.wikipedia.org/wiki/Filter_%28signal_processing%29) that passes [signals](https://en.wikipedia.org/wiki/Signal_%28electrical_engineering%29) with a [frequency](https://en.wikipedia.org/wiki/Frequency) lower than a certain [cutoff frequency](https://en.wikipedia.org/wiki/Cutoff_frequency) and [attenuates](https://en.wikipedia.org/wiki/Attenuate) signals with frequencies higher than the cutoff frequency
* מושגים נוספים:
* banpass filter
* banpass signal
* separable binomial filter
* spatial frequancy cutoff
* chrominance components
* YIQ
* IIR bandpass
* linear ramp transition
* downsampled
* quantization
* ideal banpass
* in situ
* photoplethysmogram
* revealing board
* second order IIR
* wide-sense stationary
* grey card
* feature tracking
* רגישות זמן-מרחב ("spatio-temporal sensitivity")
* סינון כפונקציה של זמן בתדירות נמוכה ("temporal pooling ")
* "quantization noise".
* ניתוח המודל המתמטי מסתמכת על קירוב לינארי של brightness constancy assumption
* optical flow.
* עבודה בגישה מערכתית (multiscale) הכוללת העצמה של תנועה ללא מעקב או קירובי תנועה.
* חלוקה לסגמנטים לפי תנועה (motion segmentation) ו- image in-painting ע"מ לקבל תמונה באיכות מספיק טובה (אשר מעלות סיבוכיות של האלגוריתם)
* Temporal processing
* localized spatial pooling
* bandpass filtering ע"מ לחלץ ולחשוף את האות המתכתב/קשור עם הדופק. אנליזת התחום הראשוני הזאת מאפשרת להעצים ולהציג את סיגנל הדופק במיקומים שונים על הפנים.
* high-pass filtering
* spatio-temporal processing אוילרי
* Scale space
* localized spatial pooling
* יישום חילוץ דופק (offline) באמצעות MATLAB
  + בנוסף למאמר, MIT סיפקו קוד במאטלב שמממש את ה- Video magnification. בשלב זה ניסינו להבין את המימוש שלהם. הרקע שלנו במאטלב לחלקנו היה קטן עד לא קיים, וזהו גם קושי שנאלצנו להתמודד איתו.
  + למדנו את הדברים הבסיסיים של השפה והתחלנו לעבור על הקוד. בקוד היה הרבה שימוש בפונקציות מוכנות בעיבוד תמונה שהיינו צריכים לקרוא עליהם ולהבין בדיוק מה הן עושות ואיך זה משתלב בקוד, בנוסף היינו צריכים ללמוד על FFT שנעשה בו שימוש.
  + לאחר שהבנו בקווים כללים את המימוש שלהם ניגשנו לשלב הבא.
  + חילוץ דופק במאטלב (offline) - כעת היינו צריכים להוסיף קוד משלנו לקוד הקיים של MIT על מנת לחלץ דופק. על מנת לעשות זאת התחלנו לקרוא הרבה מאמרים בין אם באתר הפרויקט של MIT ובין אם במקומות אחרים באינטרנט על דרכים לחילוץ דופק באמצעות video magnification. זה היה שלב של הרבה קריאה והמון ניסוי וטעייה. החלטנו ללכת על כיוון מסוים שהיה נראה יחסית פשוט ומימשנו זאת.
  + בדיעבד גילינו שניתן לקבל תוצאות טובות רק אם טווח התדרים שעבורם אנחנו עושים filtering הוא טווח בסביבה קטנה סביב הדופק. זה גם מה שהצגנו במצגת אמצע ומשלב זה החלטנו לעבור למימוש real time ב- CPP.
  + מושגים רלוונטיים אותם למדנו לצורך יישום מאטלב:
* Laplacian/Gaussian pyramid - באופן כללי פירמידה היא דרך לייצוג תמונה כאשר בכל שלב בפירמידה עושים smoothing and subsampling (וע"י כך הרזולוציה של כל שלב קטנה ונוצרת הפירמידה).
* פירמידת גאוס - לוקחים את התמונה המקורית ומפעילים עליה את Gaussian Blur (מושג המתואר בהמשך) ומה שקיבלנו זה גרסת lowpass של התמונה המקורית. ובכל שלב בפירמידה, כל פיקסל מחושב כממוצע הפיקסלים השכנים ברמה שמתחתיו בפירמידה.
* פירמידת לפלסיאן - מחושבת באופן דומה לפירמידת גאוס, רק שכל שלב בפירמידה הוא ההפרש בין התמונה המקורית לתמונה שנעשה עליה ה-lowpass filtering. התהליך הזה ממשיך על מנת לקבל קבוצה של תמונות שהם bandpass.

<https://en.wikipedia.org/wiki/Pyramid_(image_processing)#Gaussian_pyramid>

<http://www.cs.utah.edu/~arul/report/node12.html>

* lowpass filtering , highpass filtering , bandpass filtering -

כל המושגים האלו אלה אלגוריתמי "החלקה", כאשר החלקה היא שיטה למציאת תבניות ודפוסים בגרף תוך כדי הקטנת הרעשים או העלמתם לחלוטין.

lowpass מסנן תדרים שהם מעל רמת סף מסוימת שאנחנו מגדירים.

highpass מסנן תדרים שהם מתחת לרמת סף שאנחנו מגדירים.

bandpass הוא שילוב של שניהם, כלומר הוא מקבל כקלט גבול עליון וגבול תחתון, והוא משאיר רק את התדרים שבתוך הגבולות האלו.

<https://en.wikipedia.org/wiki/Passband>

* downsample - לדגום פיקסל בודד כל n פיקסלים (כלומר להשאיר רק את הפיקסלים שנמצאים במקומות שהן כפולות של n), כאשר את n אנחנו מגדירים מראש, וע"י כך לצמצם את התמונה, להפחית רעשים וכו'. דוגמאות:<http://www.mathworks.com/help/signal/ref/downsample.html>
* Gaussian Blur - זהו אלגוריתם החלקה, שעושה את ההחלקה ע"י שימוש בפונקציית גאוס.

אלגוריתם זה נחשב כlowpass filter מכיוון שהוא מסנן תדרים גבוהים.

* FFT - זאת התמרת פורייה מהירה (מימוש יעיל ל- DFT), ומשתמשים בה באלגוריתמי סינון והחלקה ( filtering and smoothing).
* הצגת התוצאות המתקבלות תוך השוואה לתוצאות הצוות המפרסם מ- MIT על סרטונים זהים
* ויזואליזציה של video magnification מאזור הפנים (וידאו)

**שלב ב'** - עד מצגת סיום

* יצירת המודל מחדש ב- ++C (זמן אמת)
  + התקנת סביבת עבודה שלקחה הרבה זמן: עבודה עם openCV, התקנה באמצעות cMake, עבודה עם QT לצורך ממשק גרפי (למרות שבסופו של דבר הפרויקט לא נבנה ב- QT), תחילת בניית קוד באקליפס, המשך בניית קוד ב- visual studio (לצורך התממשקות עם הממשק שהכנו תוך שימוש בפרויקט של Luxand כפי שיוסבר בהמשך).
  + המעבר מקוד "offline" בשלב הראשון לשלב ה- realtime השני היה מסובך - בין השאר בגלל העובדה שלא הייתה לנו אפשרות לדבג את הקוד ב- realtime (להשוות לתוצאות דופק נכונות).

מסיבה זאת הגענו לשתי החלטות:

* + - העברנו בהדרגה את הפרויקט למצב realtime אבל בעיקר התמקדנו בשיפור האלגוריתם של חילוץ הדופק ב- offline משום שכך זה קל יותר לדבג ולהשוות לתוצאות MIT. שלב זה לווה בהרבה בעיות שכללו בין היתר למידת הספרייה opencv, מציאת מימוש ל- video magnifiacation ב- cpp ועוד בעיות שנפרט בהמשך.
    - קנינו Pulse Oximeter מאושר FDA שניתן לשים על האצבע על מנת לדבג את התוצאות ב- reatime ולהשוות אותן לתוצאות הנמדדות ע"י המכשיר
  + בעיות נוספות בהן נתקלנו בשלב הזה:
    - להבין איך לממש את שני חלונות הזמן שרובינשטיין כותב עליהם בתזה (לא במאמר). חלון אחד של 15 פריימים בו נמצא בהמשך פיק/מקסימום של עצמת צבע, וחלון אחר של 5 שניות שלפיו נחשב את הדופק.
    - להתעסק עם הפריימים ברמת הפיקסלים באמצעות opencv (זה לא הלך חלק בעיקר בגלל המרות וסוגי יצוג שונים של הפריימים, שחלק נעשו בקוד של Video Magnification בו השתמשנו והיה צריך לחקור זאת).
    - להצליח לבנות נכון פונקציה שתדע לזהות רק פיקים אמיתיים של עצמת הצבע ובנוסף להצליח למצוא אותם גם בתפר שבין חלונות זמן רצופים.
    - לפתור בעיה שוידאו שצילמנו גרם לקריסת התוכנה. בדיעבד הסיבה הייתה שהתמונות בוידאו צולמו ברזולוציה גבוהה מדי כך שהתוכנה הייתה צריכה להשתמש בזיכרון מעבר לגודל הזיכרון שמערכת ההפעלה מקצה. הפתרון היה להשתמש בפונקציית resize.
    - להתחיל לבנות גם ממשק גרפי לתוכנה שישמש אותנו בהמשך לזיהוי פיקסל בודד/patch באזור מסוים בפנים בזמן אמת. כאמור ניסינו לעשות זאת באמצעות Qt Creator בעקבות המלצות שמצאנו באינטרנט אבל בסופו של דבר עשינו זאת בויזואל סטודיו.
    - דיבאגינג ארוך ושיפור התוצאות
    - חשבנו שכדי לשפר את האלגוריתם (הראשון) כדאי לבדוק ביתר פירוט את אופן תפקודה של הפונקציה שמוצאת פיקים ולנסות לשנות אותה כך שנוכל להיפטר מזיהוי לא נכון של נק' המקסימום. זאת ע"י הגדלת מס' הפריימים אליהם משווים בשביל למצוא אותן, והוספת בדיקות הסתברותיות כמו סטיית תקן או חישובים מורכבים יותר. בסופו של דבר ניסינו לגשת לבעיית חילוץ הדופק באופן אחר (האלגוריתם השני באמצעות התמרות DFT ולא זיהוי פיקים).
    - לא הבנו איך ניתן לעבוד רק עם ערוץ הצבע האדום בשלב הראשון. לאחר מכן רצינו לבדוק שאנחנו עובדים נכון עם ייצוג הפיקסלים וגם שאכן אפשר רק להתייחס לערוץ הצבע האדום. אפשרויות נוספות שבדקנו היה שימוש בכל ערוצי הצבע (RGB) בנפרד למרות שהאינטואיציה שלנו הייתה שהתוצאה של חילוץ הדופק (שקשור לזיהוי רמת האדמומיות בעור הפנים לפי המאמר של EVM) תהיה טובה יותר דרך הערוץ האדום. מאחר ובמאמר הראשי ובדוקטורט של רובינשטיין לא הייתה התייחסות לבחירת ערוצים איתם יש לעבוד היינו צריכים לחפש עוד מאמרים בספרות.
    - **בעיה מרכזית -** בשלב הראשון בחרנו לעבוד לפי ההנחיות של רובינשטיין בשיטת peak detection. הבעיה המרכזית של שימוש בשיטה זאת הייתה שהאלגוריתם לא יכול לעבוד נכון מבלי לדעת מראש טווח מצומצם מאוד של הדופק האפשרי (בטווח כזה אפילו הצוות מ- MIT השתמשו בתור הנחת יסוד, מידע מוקדם שהכניסו לקוד). כשאנחנו הגדלנו את הטווח, התוצאות המתקבלות השתנו באופן לא טוב. הסיבה לכך שהתוצאות לא היו טובות כשהגדלנו את טווח התדרים הייתה מכיוון שבעקבות הרעש נוצרו פיקים "מזוייפים" וזה הקשה עלינו מאוד לגלות דפיקות לב אמיתיות. בעיה נוספת הייתה שלא הייתה לנו דרך להשוות למשהו אחר כדי לדעת האם זה תקין (זאת אומרת האלגוריתם במאמר לא מספיק טוב) או לא (ואז יש אצלנו בעיה/בעיות שצריך לגלות ולפתור).
* יצירת אלגוריתם ראשון - וחילוץ דופק המתבסס על שיטת EVM מהמאמר + הדוקטורט של רובינשטיין מ- MIT (בשיטת peak detection)
* שימוש במצלמה חיצונית עם קצב פריימים גבוה (90fps) לשיפור בדיוק
  + מאחר והיה חוסר דיוק בתוצאות המדידה בעת שימוש באלגוריתם הראשון ניסינו לשנות כמה פרמטרים אשר עליהם לא מצאנו המלצה במאמר הראשי או בדוקטורט של רובינשטיין אך נראה כמו נסיון ששווה לעשות. נעשה שימוש במצלמת PT-Gray אשר לה קצב פריימים גבוה פי 3 ממצלמת הרשת בלפטופים שלנו (עד 90 fps) והיא גם איכותית יותר. הדבר לא השפיע על תוצאות המדידה. לאחר התייעצות עם ד"ר אסף גלזר מ-nanit הוא המליץ לנו לחזור ולהשתמש במצלמת 30 fps סטנדרטית.
  + נעשה ניסוי עם מצלמת ה-PT-Gray ועם מצלמות נוספות שהיו זמינות עבורנו (בהמלצת המנחה רון - שימוש במצלמה של אייפון + שימוש בפונקציית ה- slow motion על מנת לקבל מספר גבוה יותר של פריימים בשניה ובנוסף רזולוציה גבוהה יותר). גם במקרה הזה לא ראינו הבדל גדול בתוצאות כאשר בשלב זה הבעיה העיקרית שלנו הייתה חוסר יציבות שלהם (בבדיקה אחת הערכים נעים ב +-20 bpm לכל כיוון).
* יצירת אלגוריתם שני - וחילץ דופק המתבסס על שיטת EVM והתמרות פוריה (DFT) למציאת התדר המתאים לדופק הקיים של ה"פציינט".
  + במאמרים המקוריים איתם עבדנו הייתה המלצה להעדפת שיטת ה- peak detection על פני שיטות אחרות. המלצה זאת הופיעה בדוקטורט של רובינשטיין (מאמר מס' 1) ובמאמר של הצוות מאיראן (מספר 4). לאחר התייעצות עם המנחים רון וגיל החלטנו לנסות ליישם שיטה שניה שבה החלפנו את שיטת ה- peak detection שנתנה תוצאות לא יציבות, בהתמרת פוריה מסוג (DFT (FFT. בנוסף מצאנו במאמר (מס' 3) שצוות מגרמניה שיישם את שיטה זאת פרסם שהיא עובדת בצורה יציבה ומספקת תוצאות איכותיות.
* יצירת Patch מתאים
  + לפי המאמרים השונים אותם קראנו קיימות שיטות שונות לבחירת הפאטצ' המתאים אשר כאשר נעבוד רק לפיו (לחתוך את הפריים רק לאזור העניין לפי הפאטצ' שרוצים) - תתקבל הקורלציה הגבוהה ביותר לדופק המתקבל בבדיקה עם מכשיר FDA סטנדרטי.
  + Xiaobai Li, Jie Chen (מאמר מס' 5) - טענו שהתוצאה האיכותית ביותר מתקבלת ע"י יצירת patch מאזור הסנטר (בצורת "חצי עיגול")
  + Matt Estrada והצוות מאוניברסיטת סטנפורד (מאמר מס' 6) - טענו שהתוצאה הטובה ביותר מתקבלת ע"י יצרת patch מאזור הסנטר (בצורת "חצי עיגול")
  + Hao-Yu Wu והצוות מ-MIT (מאמר מס' 7) - טענו שהתוצאות הטובות ביותר מתקבלות ע"י יצירת patch מאזור הלחיים והמצח. צוות זה הגדיל לעשות ויצר פונקציה אשר יצרה patch מורכב שהתקבל ע"י "משקול" האזורים הרלוונטיים ביותר לחילוץ דופק מאזור הפנים - כך שה- patch שהתקבל לא היה בצורה גיאומטרית (מלבן/חצי עיגול וכו') אלא בצורה של אוסף נקודות בעלות משקל שונה כך שנקודה שקיבלה משקל גבוה תשפיע יותר על הממוצע המשוקלל של חישוב הדופק הסופי.
  + בהתבסס על הכלים והזמן אשר עמד לרשותנו החלטנו להתמקד ב- patches אשר ידענו כי נוכל לבצע (צורות גיאומטריות פשוטות).
  + על מנת להתחיל בתהליך בניית ה- patches היינו קודם כל מוכרחים להטמיע אלגוריתם ויישום של face detection קיים. בהתחלה ניסינו לעבוד עם OpenCV ועם דוגמאות שונות שחיפשנו ב- GitHub. רוב התוצאות של החיפוש לא עבדו בצורה מספיק מדויקת ולכן התחלנו לעבוד עם SDK -ים שונים אשר מכילים בתוכם פונקציות בנושא face-detection. בסופו של דבר הצלחנו ללמוד איך לעבוד עם FaceSDK של Luxand ומשם התחלנו לבנות את הממשק הגרפי כולל face-detection ו- face-recognition.
  + כל עוד אנחנו ממשיכים לעבוד עם patches גאומטריים מרובעים או מלבניים, יכולנו ליצור patches שונים על כל הפנים, על אזור הלחיים, אזור המצח ואזור הסנטר.
  + בשביל להבין איך להתקין ולעבוד עם Luxand היה עלינו לקרוא מידע בקישור:

<http://luxand.com/facesdk/>

ובין השאר, מלבד קריאת מידע ושינוי של הפונקציות בהן עושים שימוש שם, גם להבין את השליטה בממשק הגרפי שעושה שימוש ב- DialogBox של windows כפי שמופיע בקישור:

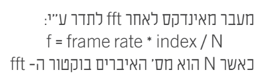
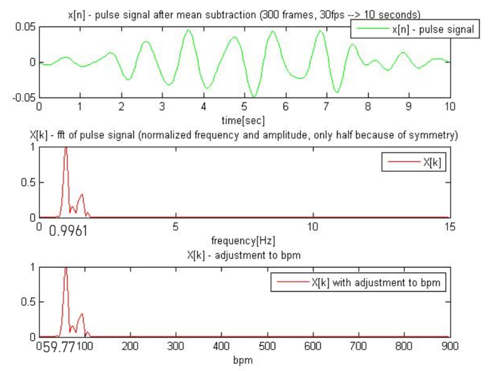
<https://msdn.microsoft.com/en-us/library/windows/desktop/ms632588(v=vs.85).aspx>

* + נתקלנו במס' בעיות כאשר הגענו לנקודה בה יש לשלב בין חישוב הדופק אל הממשק הגרפי.
    - קודם כל היה צריך ליצור מחדש פרויקט ולהשתמש במדריך של Luxand לאופן הגדרות הפרויקט.
    - גם ב- Luxand משתמשים במצלמת אינטרנט וגם בקוד הפרויקט ב- opencv על מנת לקבל את הפריימים בזמן אמת. זה הוביל להתנגשות ולכן בסופו של דבר הפרויקט עובד כך שהשימוש במצלמה נעשה רק ב- opencv והתמונות נשלחות אל - Luxand.
    - לכן גם בסופו של דבר הפאטצ' בו השתמשנו נוצר ע"י שימוש ב- face-detection של opencv (למרות שיש שימוש ב- face-detection גם ב- Luxand).
    - ראינו שהממשק הגרפי נתקע כל פעם כאשר מנסים לחשב את הדופק למשך כשתי שניות. לכן לא הייתה ברירה אלא לשנות את הקוד כך שהוא יעבוד באופן מקבילי. יצרנו שני threads לשם כך: הראשון עבור חישוב הדופק והשני עבור הצגה ויזואלית של הממשק.
* חיבור כל האלגוריתמים והפיצ'רים לכדי תוכנה אחת עובדת
  + בזמן יישום חילוץ הדופק בזמן אמת ממקטע הוידאו בזמן אמת היה צורך לייצר buffer של 150 פריימים אשר ייצגו 5 שניות של צילום פרונטלי של הפנים אשר מהם נרצה לחלץ את הדופק בקצב של 30fps. לאחר הפעלת שיטת EVM ו- temporal bandpass filter נקבל את אותו מקטע וידאו עם תדרי הדופק הרצויים מוגברים, המלצת המאמר בטווח Hz0.83-2.5 (שזה מתקשר לדופק בין 50 ל-150 פעימות בדקה, BPM). באמצעות קטע הוידאו הזה (נקרא לו בשם "filtered video") נוכל להתחיל לחלץ את קצב הדופק בדרכים שונות. בשלב הראשון נמצע את עצמת הצבעים בכל פריים ונייצר וקטור בעל 150 ערכים המייצגים את ממוצע ה-intensity בכל נקודה בזמן מה-ROI (region of interest) הנבחר, במקרה שלנו patch מאזור המצח. הווקטור אותו יצרנו (נקרא לו בשם "pulse vector") מכיל 150 ערכים בהם נמצא ממוצע הערכים הרצויים בכל נקודה בזמן ולמעשה מכיל את המידע על קצב הדופק של הנבדק.

כעת ניתן להשתמש במספר שיטות על מנת לחלץ מה- pulse vector את הדופק בזמן אמת.

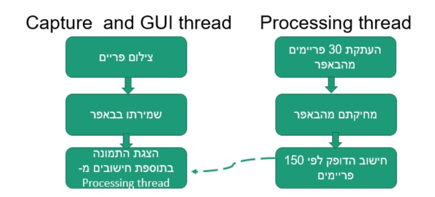
השיטה הראשונה היא שיטת peak detection אשר תוארה בצורה נרחבת בחלק הראשון של ספר הפרויקט. באמצעות שיטה זאת שהוצגה בתזה של מיכאל רובינשטיין מ- MIT ניתן להשתמש ב- "pulse vector" ולזהות מתוכו את ה- peaks השונים (בתוספת אופטימיזציה לזיהוי הפיקים תוך שימוש בממוצע קטום) ולאחר מכן להשתמש בנוסחה 60 \*numOfPeaks / WindowsSizeInSeconds וכך נקבל את הדופק בפעימות לדקה.

השיטה השנייה לחילוץ דופק מה- "pulse vector" היא שיטת התמרות DFT שתוארה גם כן בחלק הראשון של ספר הפרויקט. בשלב הראשון נבצע התמרה של ה- "pulse vector" למישור התדר (לאחר ריפוד באפסים לגודל אופטימלי) ומשם נוכל לחשב את הערך המוחלט (magnitude). לאחר ביצוע פעולה זאת המספר המקסימלי יתקבל באינדקס אשר מייצג את תדר הדופק. הדופק המתקבל הוא ביחידות Hz, את ההמרה לפעימות בדקה bpm נעשה ע"י הכפלת התוצאה ב- 60. כדי לקבל תוצאות מדויקות יותר נשתמש בשתי טכניקות: 1. נשתמש בחלון Hamming לפני ביצוע ההתמרה (מעשית כופלים את ערכי ה-pulse vector בחלון ה-(Hamming . 2. נפחית את רכיב ה- DC מה- "pulse vector" ע"י הורדת הממוצע של הערכים בווקטור. אחת מהבעיות במהלך הפרויקט הייתה שתוצאות חילוץ הדופק לא היו מספיק מדויקות. לאחר נסיונות רבים לפתור זאת גילינו שהשימוש בהפחתת ה- DC שיפר משמעותית את התוצאות (הסבר לכך ניתן למצוא בקישור הבא: <http://blog.originlab.com/data-handling/how-to-remove-dc-offset-before-performing-fft>)



* הרחבה והסבר המימוש בנוגע לעדכון הדופק בזמן אמת

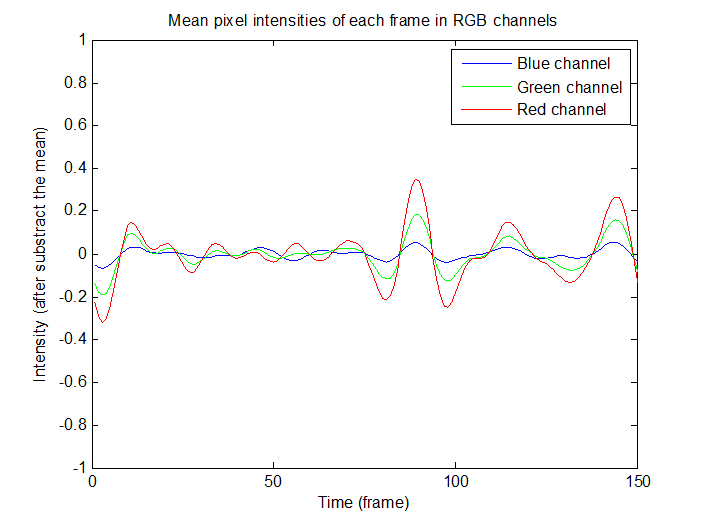
הראינו קודם לכן איך ניתן לחלץ בזמן אמת את הדופק מתוך ה- "pulse vector" **שלנו: על מנת לעשות זאת היה צורך לפתח מנגנון אשר יאפשר לעדכן את ה-**pulsevector **בקצב מהיר מספיק אשר יוכל לתת חיווי על הדופק המשתנה של הנבדק. החלטנו כי עדכון ה-** pulse vector **כל שניה 1 יהיה הפתרון האופטימלי (בקצב של** 30fps **מדובר על 30 פריימים חדשים), כאשר בעת עדכון ה-**pulse vector **נמחק ממנו את 30 הפריימים הישנים ביותר ונוסיף לו את 30 הפריימים אשר התקבלו בשניה החולפת. בדרך זאת המשכנו לפעול לאורך כל הוידאו וקיבלנו תוצאות אשר התעדכנו באופן רציף. במהלך היישום בזמן אמת נתקלנו בקשיים רבים: כל עוד התוכנית מומשה באופן סדרתי זמני החישוב הארוכים של הדופק גרמו לעצירה/האטה של התוכנית ויצרו הטיה של החישוב, ובנוסף פריימים רבים מהמצלמה היו הולכים לאיבוד ולכן לא ניתן היה לקבל תוצאות מדויקות. הפתרון לכך היה ליצור תוכנית מקבילית תוך שימוש בשני threads ובאפר של פריימים באופן הבא:**



ה- thread השמאלי ("capture and GUI thread") רץ כל הזמן ושומר בבאפר פריימים מהמצלמה.

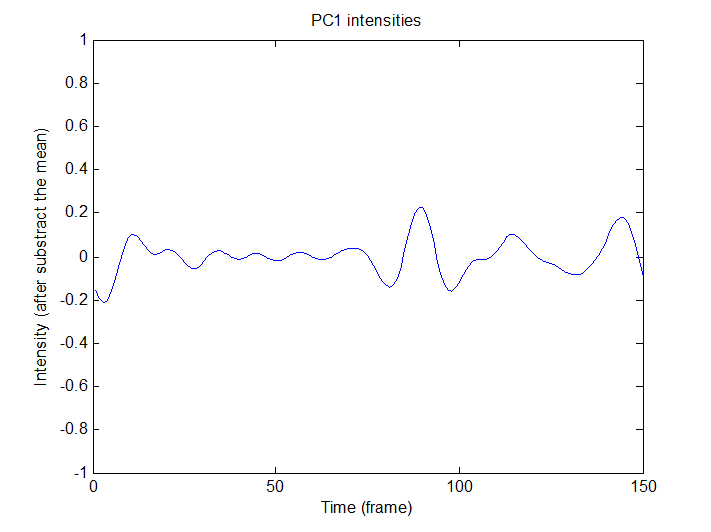
ה- thread הימני ("processing thread") מבצע את חישוב הדופק כאשר נכנסים 30 פריימים חדשים ע"י ה- thread השמאלי (רק כאשר כבר יש ברשותו 150 פריימים). כדי לעשות זאת הוא מעתיק אותם לבאפר שלו (באפר אחר שיכול להכיל עד 150 פריימים), מוחק אותם מבאפר הפריימים של "capture and GUI thread" כדי שבאפר זה יוכל להתמלא מחדש ולבסוף מחשב את הדופק לפי הבאפר בו שמורים 150 פריימים (כל פעם מוחקים ממנו את 30 הפריימים הישנים ומוסיפים את 30 הפריימים החדשים).

* שימוש ב- PCA לשיפור רמת הדיוק - המלצה של המנחים גיל ורון אשר גם מיושמת במס' מאמרים שעוסקים בחילוץ דופק
  + במאמרים שונים קראנו על טענות שונות הנוגעות לבחירת הערוץ המתאים ביותר לדגימת הדופק.
  + הופתענו לגלות שבמאמר הראשי (מס' 1) ובדוקטורט של רובינשטיין לא דובר כלל על בחירת ערוץ, בפרט על בחירת הערוץ האדום שנראה כמו הבחירה הכי אינטואיטיבית (חילוץ הדופק מתאפשר בעקבות מחזור זרימת הדם באזור הפנים).
  + מעבר לכך במאמר של Xiaobai Li, Jie Chen (מאמר מס' 5) ובמאמר של הצוות מאוניברסיטת סטנפורד (מאמר מס' 6) - הומלץ להשתמש בערוץ green בלבד - הטענה הייתה ששימוש בו מביא לקורלציה הכי גבוהה בתוצאות הדופק בהשוואה לדופק האמיתי.
  + תחילה נציג את ההבדלים של השתנות עצמת ערוצי RGB לאורך הזמן (בגרף מוצגים ערכי RGB של ממוצע הפיקסלים בפריים עבור 150 פריימים, אלה נתונים של אדם שצולם בוידאו במצלמת אינטרנט של מחשב נייד באמצעות הרצת הפרויקט בזמן אמת):

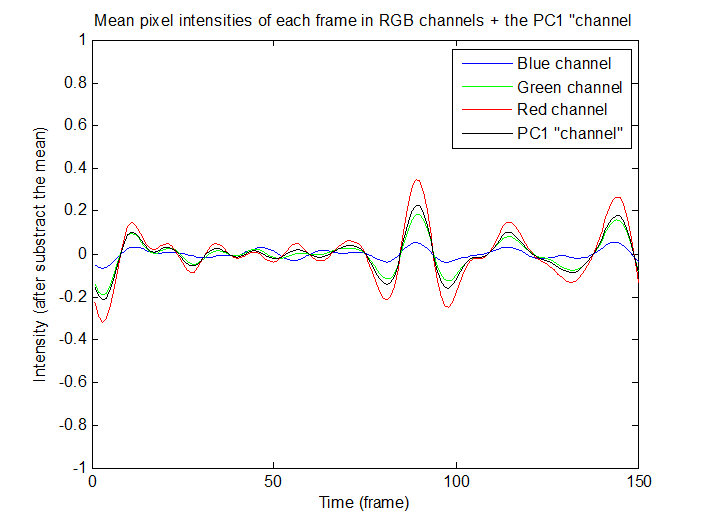


* + לאחר הנחייה מגיל ורון, ביצענו בקוד בזמן אמת (++C) וגם ב- Matlab שימוש ב-PCA (ע"י פירוק SVD) על מנת למצוא את שילוב הערכים של שלושת הערוצים (RGB) אשר ייתן את הקורלציה הכי גבוהה לדופק האמיתי. השילוב שהתקבל (first principal component) מוצג בגרף הבא (בו יש טעות שתובהר בהמשך).

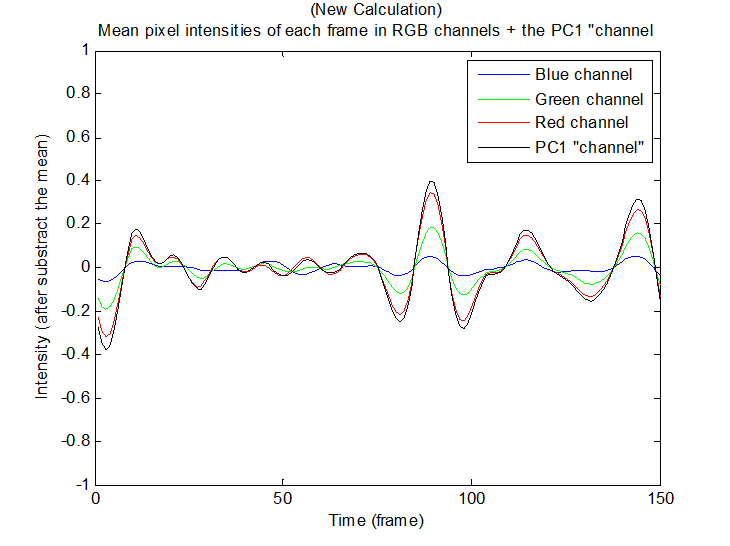
השימוש ב- PCA מאפשר למצוא את הציר המרכזי שלאורכו הערכים משתנים הכי הרבה (שונות הכי גדולה) ובכך בין השאר לסנן רעשים ולקבל מידע משוקלל של כל הערוצים.



* + בגרף הבא ערוץ משוקלל זה מוצג לצד הגרף הקודם (להשוואה נוחה יותר):

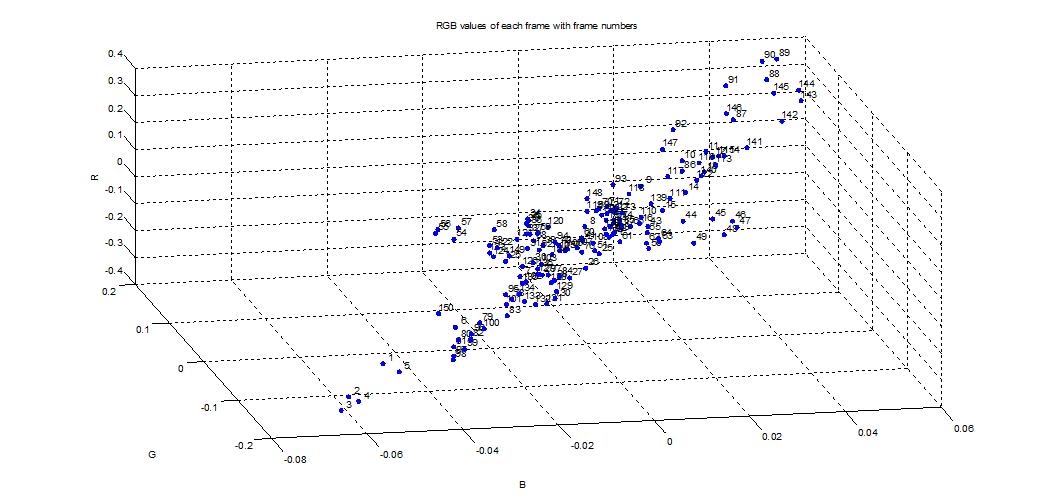


* + ניתן לראות עפ"י ניתוח עקום ה- PCA והעקומים האחרים של הערוצים השונים (RGB) כי המתאם הכי גבוה קיים בין ערוץ ה- Green ובין ה- PCA מה שמאשש את הטענה של הצוותים ממאמר מס' 5 ומאמר מס' 6 (בהנחה ששימוש ב- PCA אכן משפר את התוצאות כפי שכתוב במס' מאמרים).
  + לאחר קבלת התוצאות והצגתם למנחים גיל ורון, הם הפנו אותנו לשים לב שהיה עלינו להשתמש באותם הנתונים אך בסדר הפוך בין השורות והעמודות, כלומר להשתמש ב- transpose של מטריצת הנתונים (להשתמש במטריצה בגודל nx3 במקום 3xn כאשר אצלנו 150n = ). לכן הוטענו לחשוב קודם שהקורלציה הכי גבוהה היא בין הערוץ הירוק ל- PCA. בחישוב החדש, כפי שנראה בהמשך, דווקא הערוץ האדום הוא בעל הקורלציה הגבוהה ביותר ל- PCA:

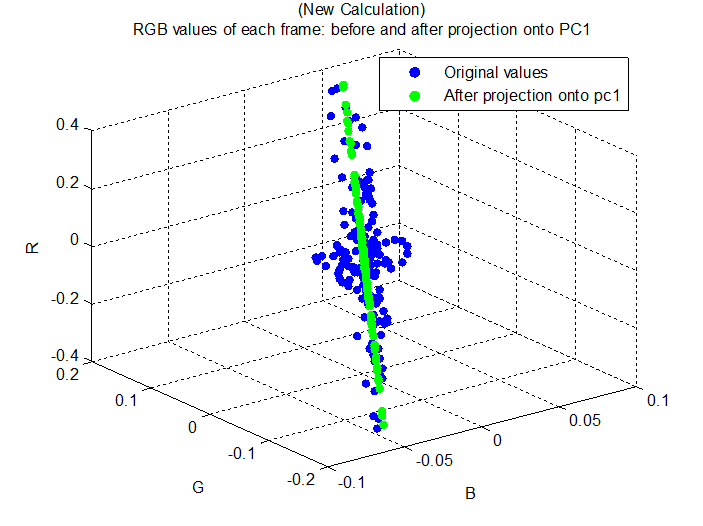


* + בנוסף רון וגיל המליצו לנו לעבוד עם היטלי הנקודות שנדגמו, שמוצגות ב"ענן" F מימדי של נקודות RGB בגרפים הבאים.

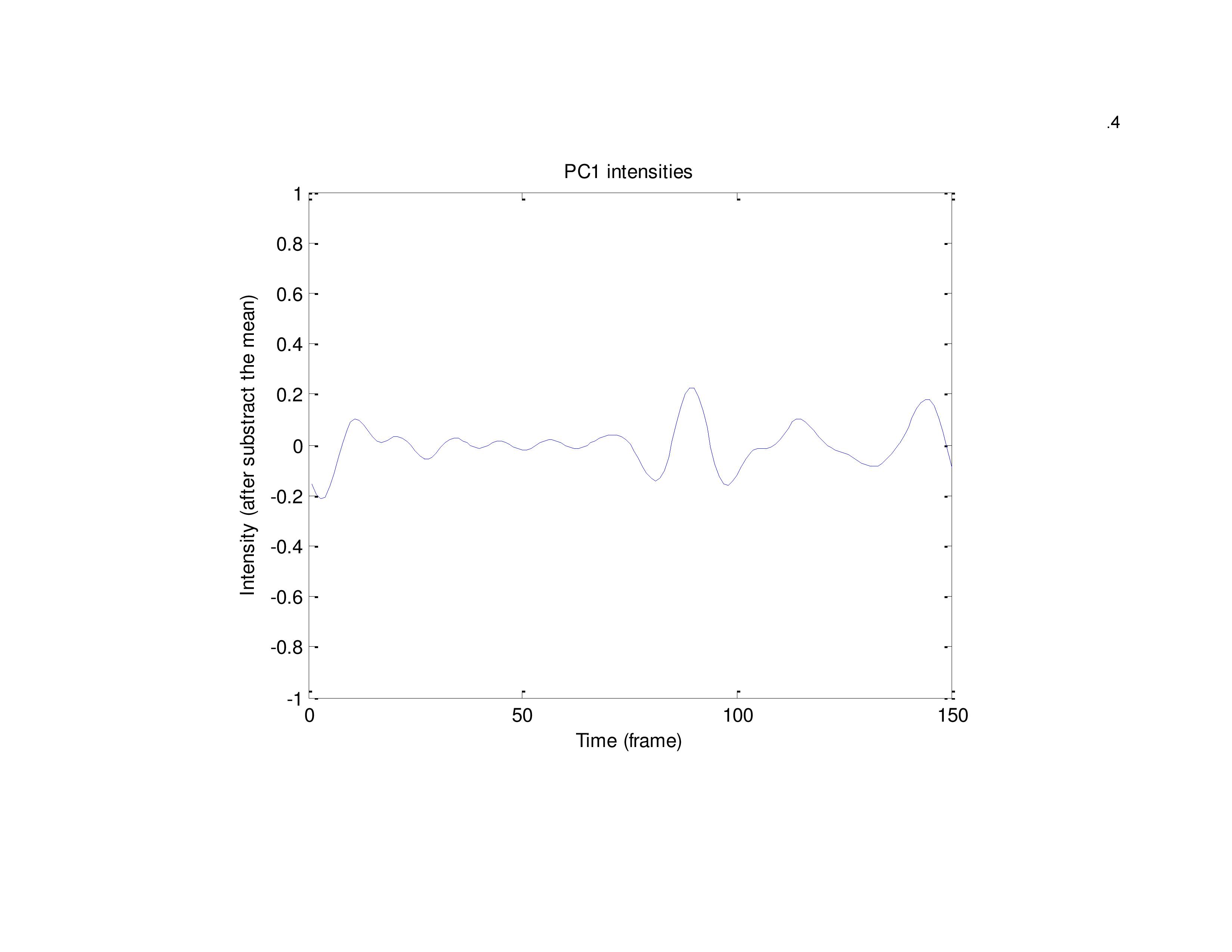
בגרף הראשון רשומים גם מספרי הפריימים מעל כל נקודה (המייצגים את הזמן - 150 פריימים נדגמו בקצב של 30 fps ולכן סה"כ הנתונים התקבלו ב- 5 שניות):



* בגרף השני אלה אותן הנקודות (מזווית אחרת) כאשר הנקודות המסומנות בכחול הן אותן הנקודות בדיוק, והנקודות המסומנות בירוק אלה אותן הנקודות לאחר ההטלה בכיוון של first principal component - pc1:



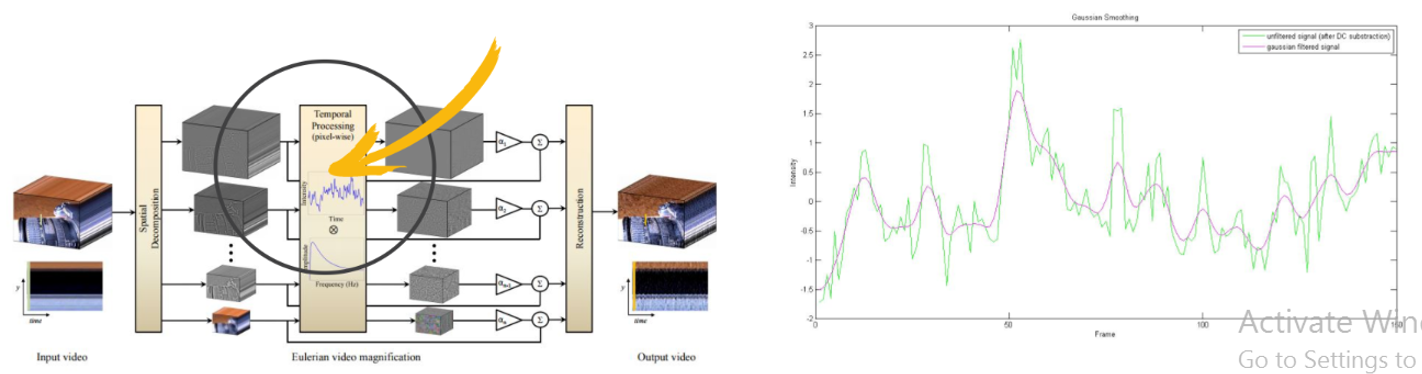
* + עבודה עם היטלי הנקודות על הציר המרכזי שקיבלנו לאחר הפעלת PCA נותנת את האפשרות לעבוד מחדש על ישר עם נקודות דגימה בשני מימדים ולבצע את החישובים על הנקודות שלפיו כמוצג בגרף הבא:



מחקר על שיטות נוספות

**1.** החלפת מסנן תדרים - **לפי המאמר את ה-**temporal processing  **יש לעשות באמצעות מסנן תדרים אידאלי כפי שכתוב במאמר מ-**MIT  **(כאשר מטרת ה-**temporal processing **היא לסנן רק את שינויי הצבע שרלוונטיים לתדרי הדופק המתאימים). בדקנו את השינוי כתוצאה מהחלפת מסנן זה למסנן גאוסיאני. המסנן פועל בצורה כזאת שהוא מחליק את האות בזמן ולכן מוריד תדרים גבוהים. בפועל החלפת המסנן למסנן גאוסיאני הובילה לתוצאות פחות טובות.**

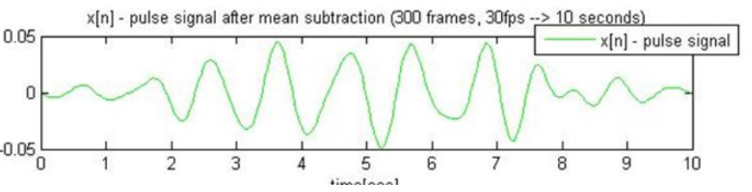
**בתמונה הבאה מתוארת אופן פעולתו של מסנן גאוסיאני לדוגמה:**



**2.** חישוב דופק בשיטה נוספת – הפרשי וקטורי סיגנל ו- **L2 norm.**

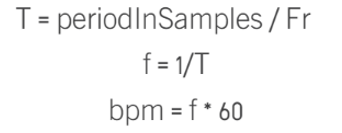
**הסבר קצר על השיטה:**

**הסיגנל של שינויי עוצמת צבע העור הוא מחזורי, למשל:**



**אנחנו מעוניינים למצוא את המחזור T משום שהוא מעיד באופן ישיר על התדר f = 1/T.**

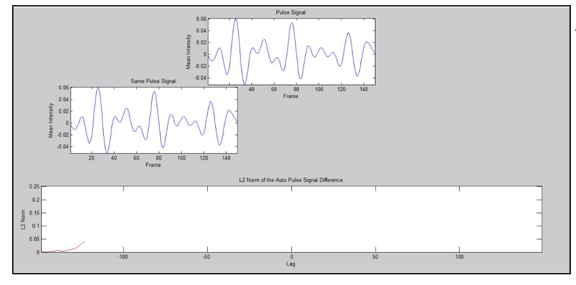
**לאחר שנתחשב גם ב- frame rate Fr = נקבל את קצב הדופק:**



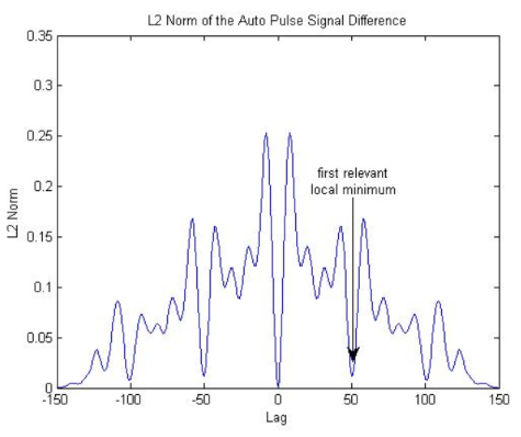
**כדי לעשות זאת נזיז את הסיגנל בזמן ב- lag = 1 ונחסר את סיגנל זה מהסיגנל המקורי ללא ההזזה.**

**כעת נחשב L2 Norm לסיגנל ההפרש.**

**נפעל באותו האופן עבור הזזה בזמן, הפעם ב- lag = 2, ואח"כ גם עבור lag = 3 וכן הלאה.**

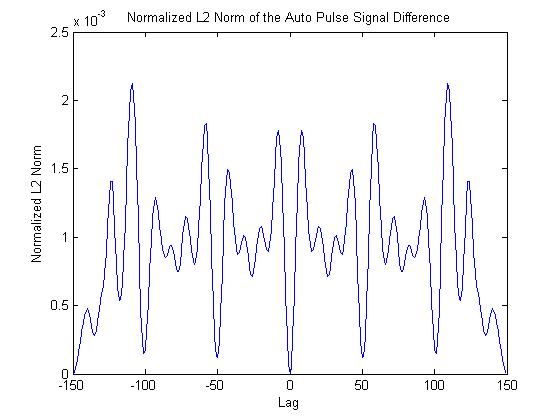


**נשים לב שככל שהבדל הזמנים מתקרב למחזור האמיתי של האות, כך ה- L2 Norm צפוי להיות נמוך יותר. לכן נחפש את המינימום הראשון שנמוך מספיק ומתאים למחזור הדופק. מינימום זה הוא הערכתנו למחזור הדופק ולכן ממנו נוכל לקבוע את קצב הדופק.**



* **המנחים הפנו אותנו לשים לב לכך שהתוצאה לא אמורה לדעוך ל- 0, ושכדי לפתור זאת יש לנרמל באמצעות חלוקה במס' האיברים בסיגנל ההפרש (כאשר מחשבים את סיגנל L2 Norm).**

**לאחר הוספת שינוי זה במימוש התמונה הקודמת נראית כך:**

****

first relevant

local minimum

* **לצורך מזעור שגיאות ושיפור דיוק החישוב נפעל באותו האופן עבור כל חלק של הסיגנל. כלומר נחשב זאת על הפעם מהדגימה השנייה שלו ועד סופו, אח"כ מהשלישית, רביעית וכן הלאה.**
* **כך נקבל עבור כל הרצה של האלגוריתם את קצב הדופק.**
* **לבסוף נחשב ממוצע כדי להכריע מהו קצב הדופק הסביר ביותר.**

1. . <https://www.degruyter.com/view/j/cdbme.2015.1.issue-1/cdbme-2015-0018/cdbme-2015-0018.pdf> [↑](#footnote-ref-1)