**דו"ח התקדמות – פרוייקט הנדסה 2014-15**

כאן יוכנסו תמונות מהרצת הSAMPLE

30\4\2014

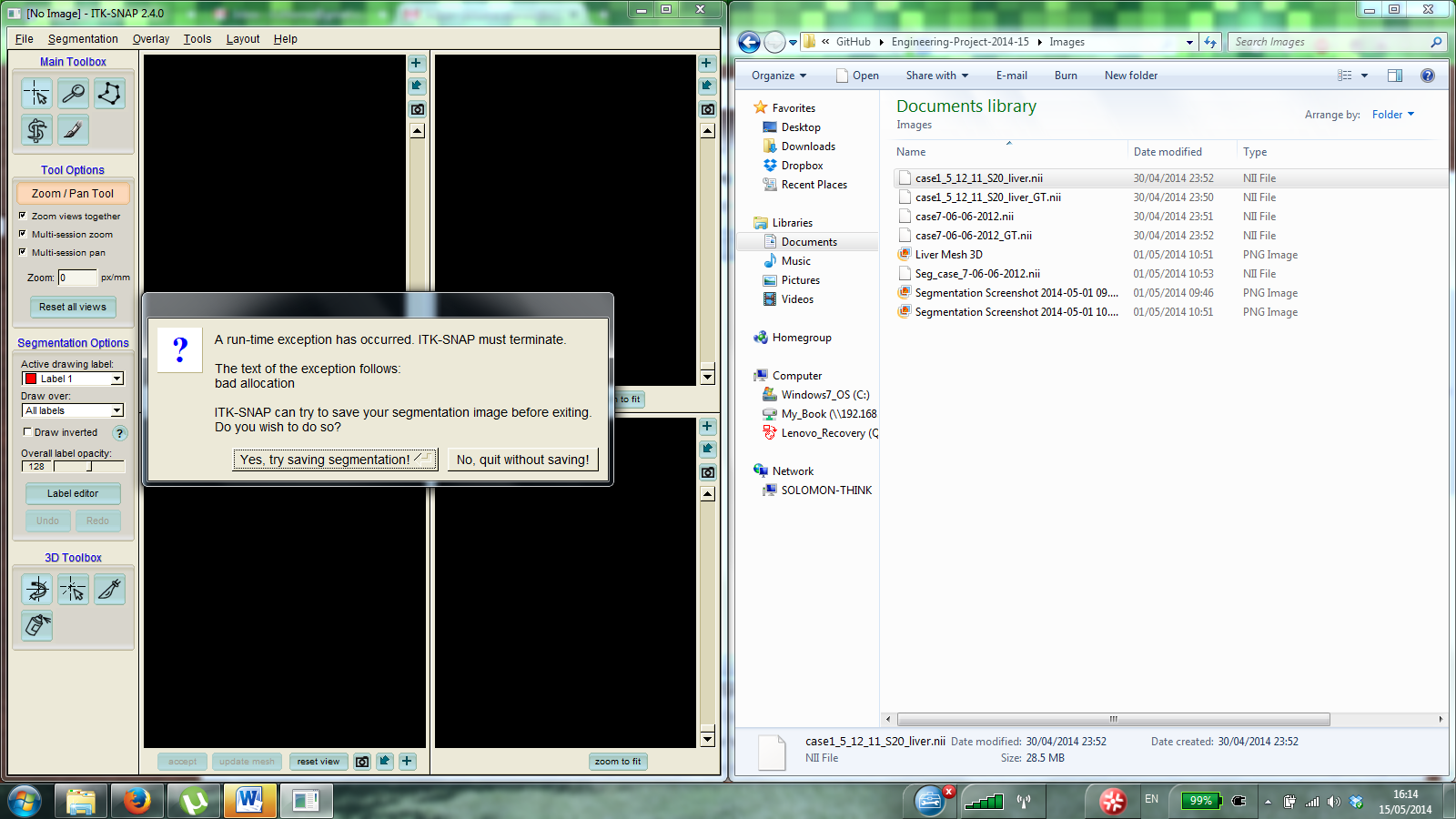
ניסינו להפעיל את ITK.

ליתנו לנו שתי תמונות לסגמנטציה:

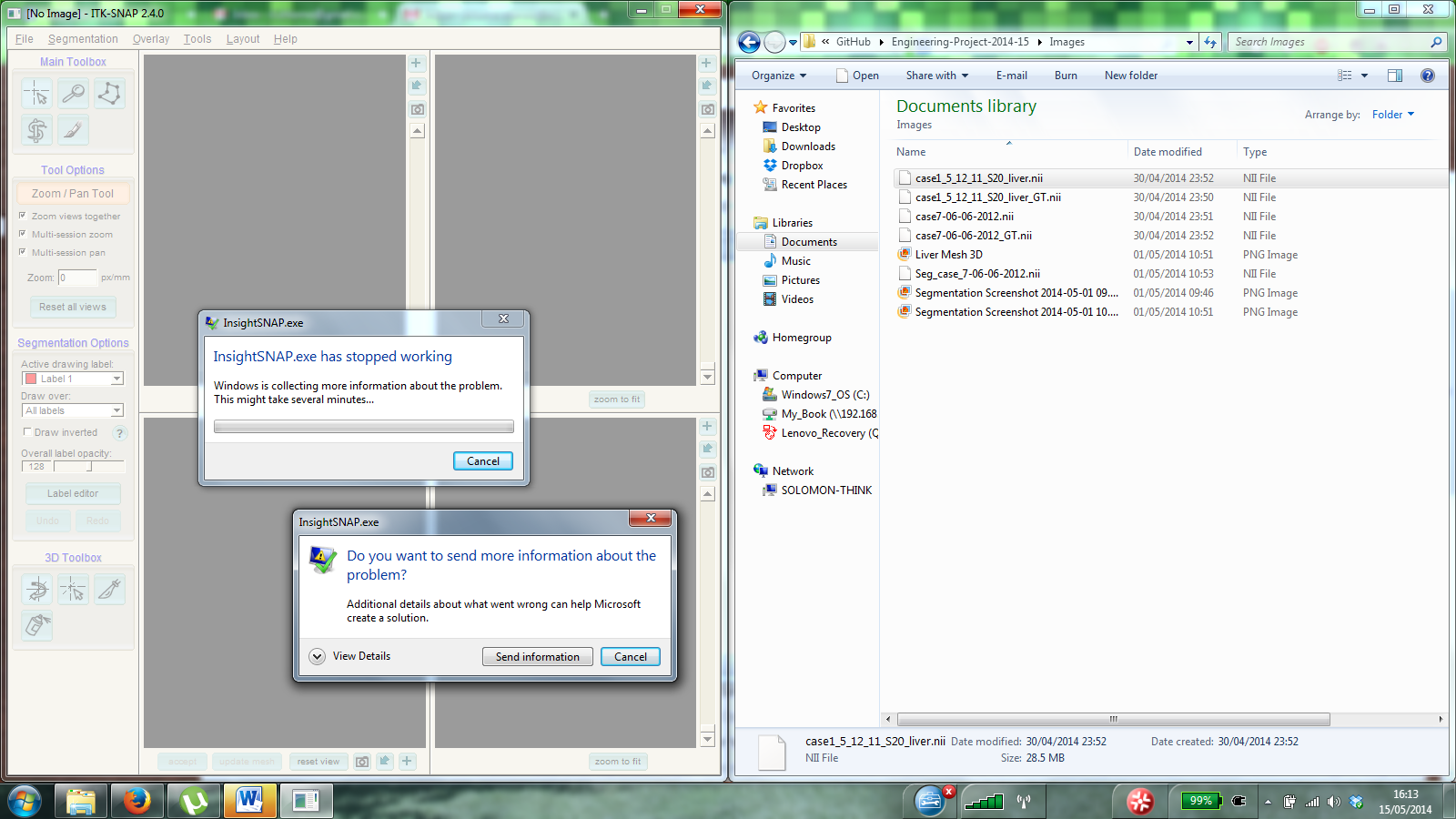
1. case1\_5\_12\_11\_S20\_liver.nii
2. case7-06-06-2012.nii

וכן ה GROUND TRUTH שלהן.

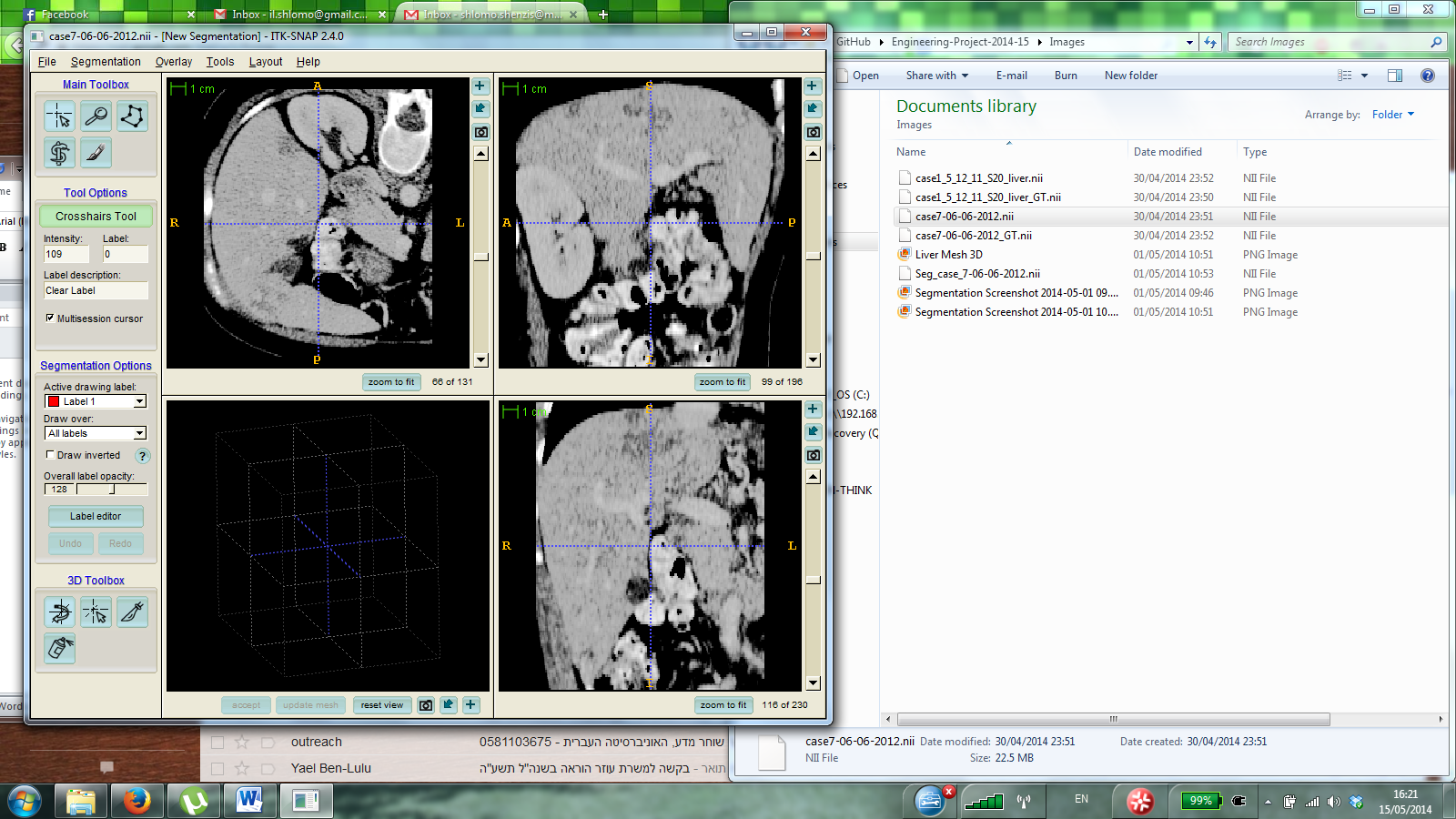
עבור תמונה 3, נכשלה הפתיחה בITK עם הטעות הבאה:



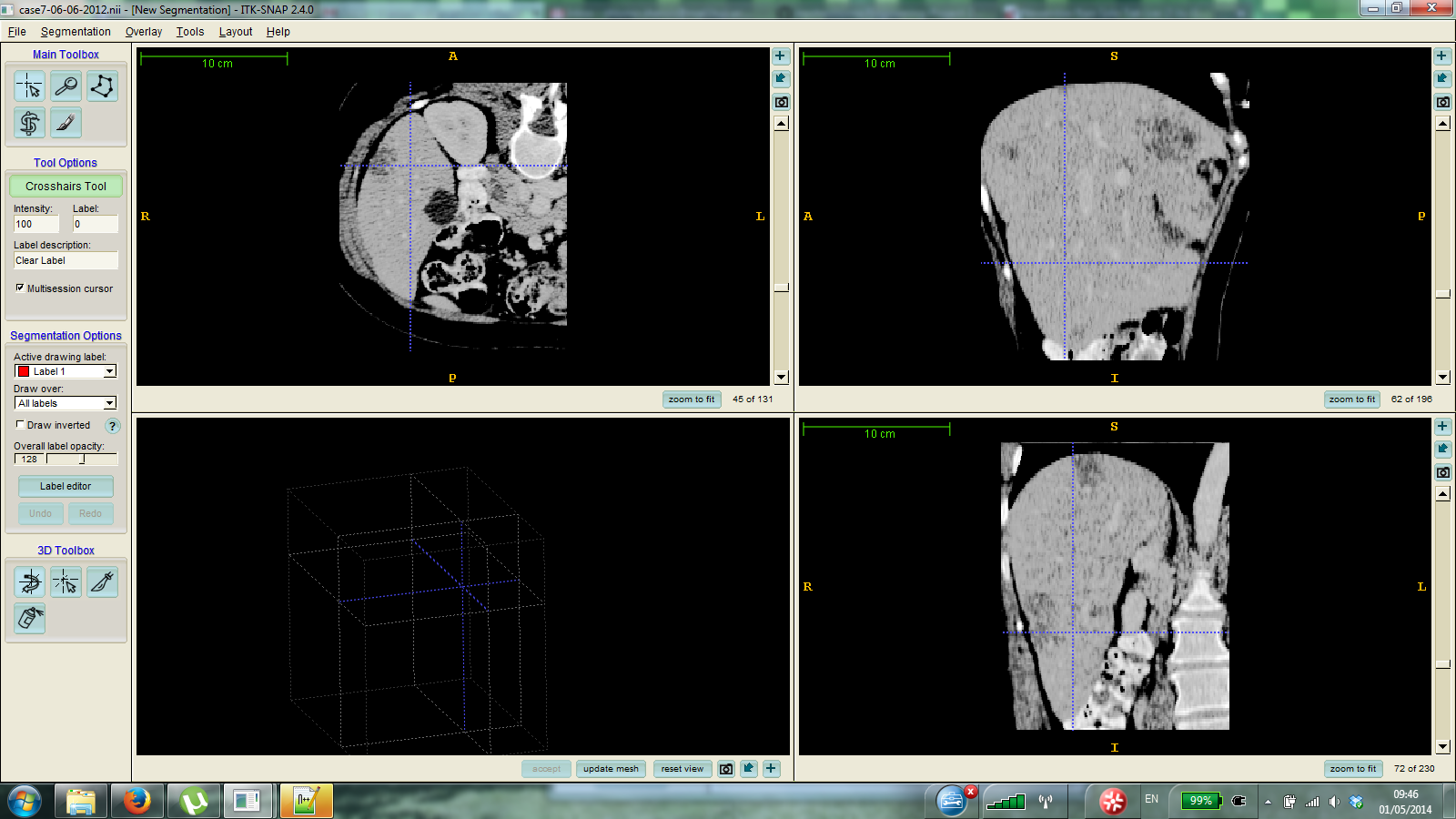
ואם ניסינו להכניסה בעזרת DRAG & DROP:



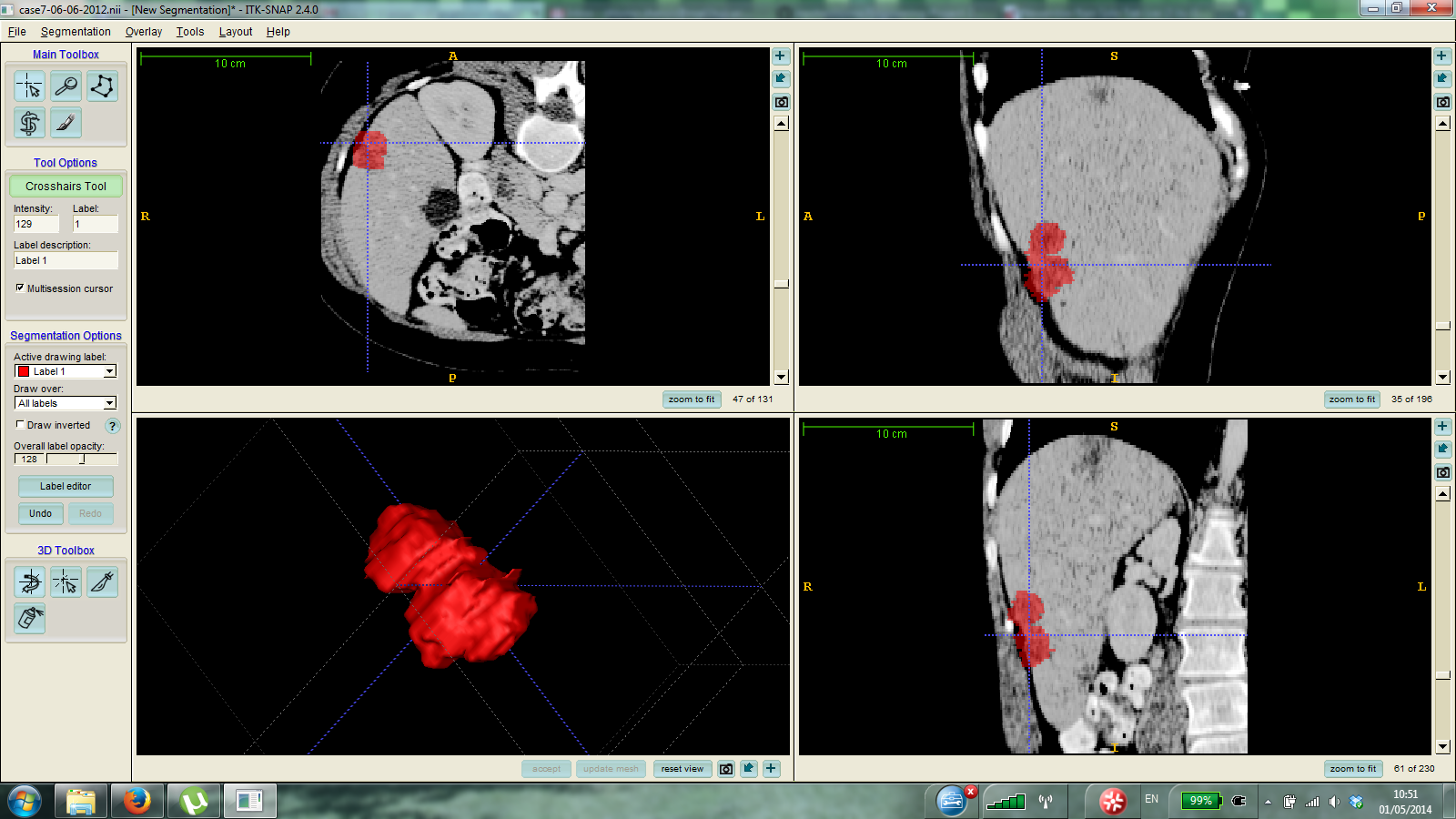
עבור הCASE השני הצלחנו לפתוח ע"י DRAG&DROP:



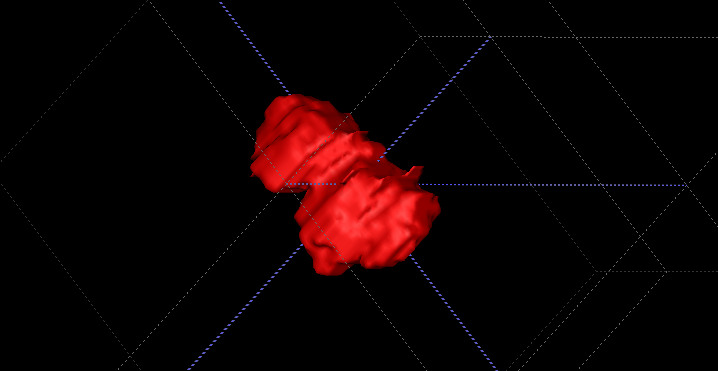
שיחקנו עם הקונטרסט:



ועשינו סגמנטציה ע"י סימון הגידול:



הMESH שיצא לנו נראה כך:



כמו כן, פתחנו GIT ב GITHUB מכיוון שSOURCEFORGE לא היה נוח מספיק.

הURL לGITHUB הינו:

<https://github.com/mosheshlomo/Engineering-Project-2014-15>

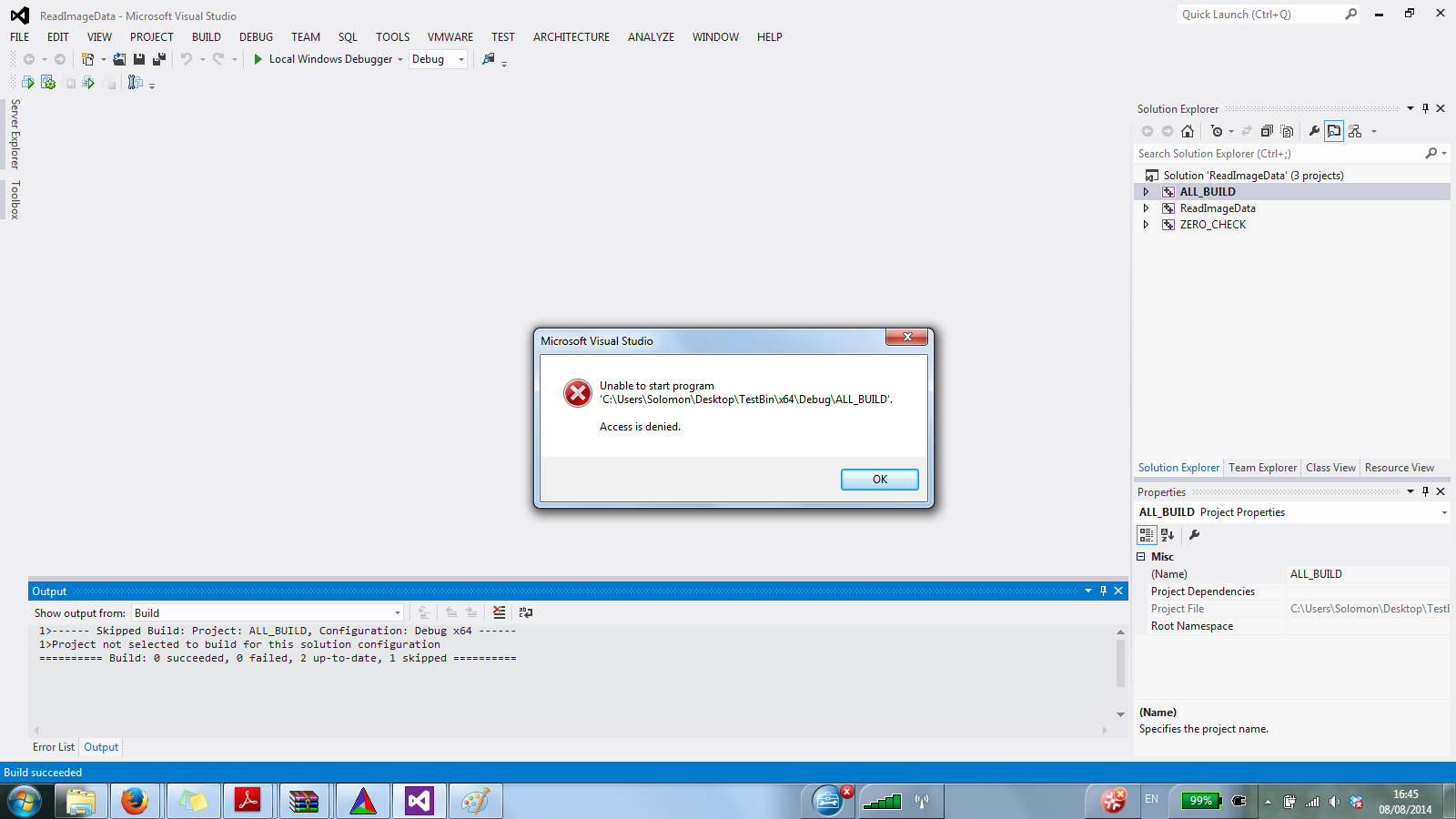
**07.08.2014**

הותקן VTK על מחשב פרטי ו גרסאת הVTK PYTHON על מחשב המעבדה ע"י טניה.

גרסא 6.2 מהGIT של KITWARE.

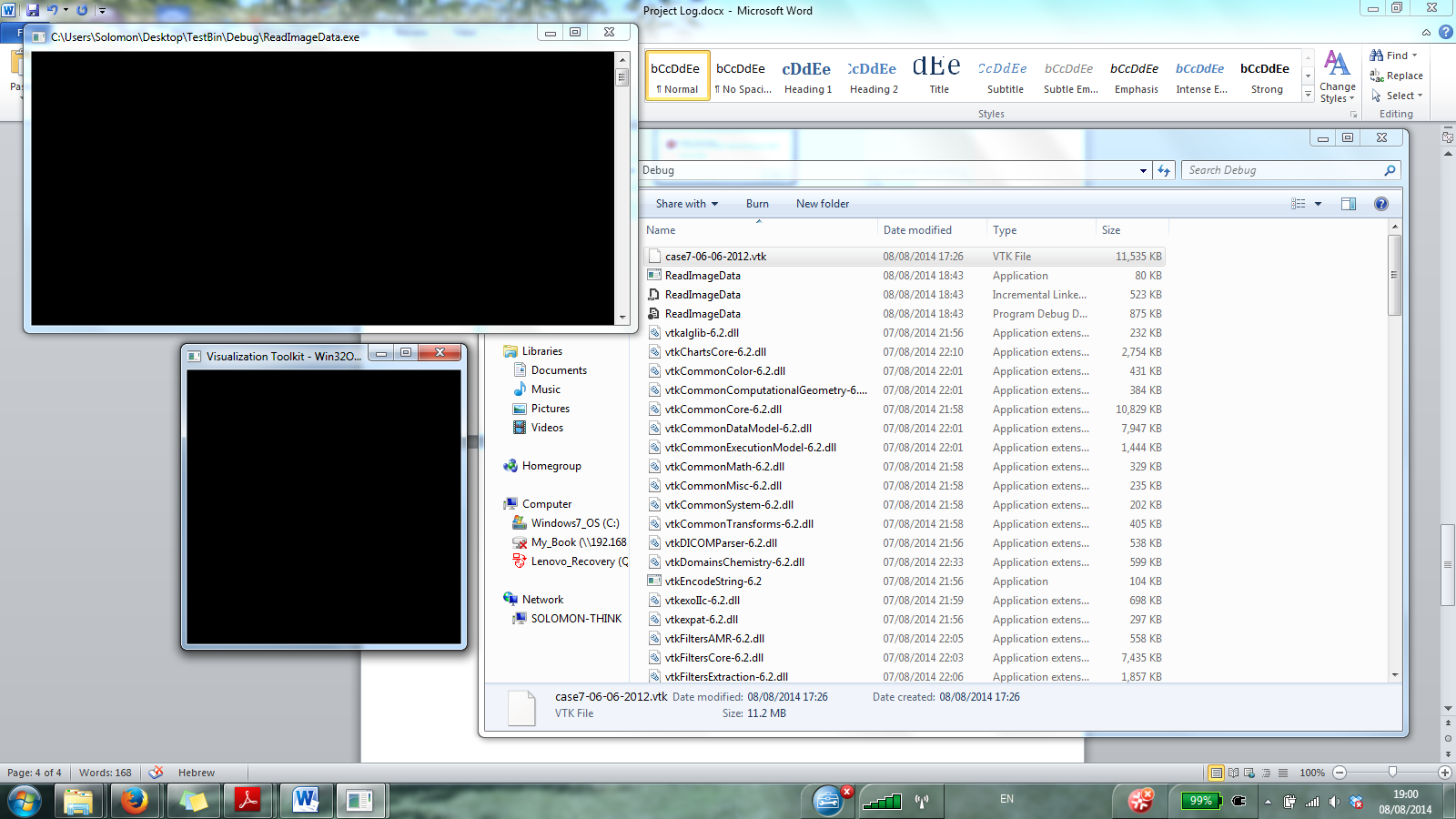
**08.08.2014**

ניסיון לבנות EXAMPLE של VTK שמראה SLICE של CT . הבניה בCMAKE וויסואל סטודיו 2012 מצליחה, אבל הרצה מתוך VS אינה פועלת:



התמונות גם בקובץ (Images).

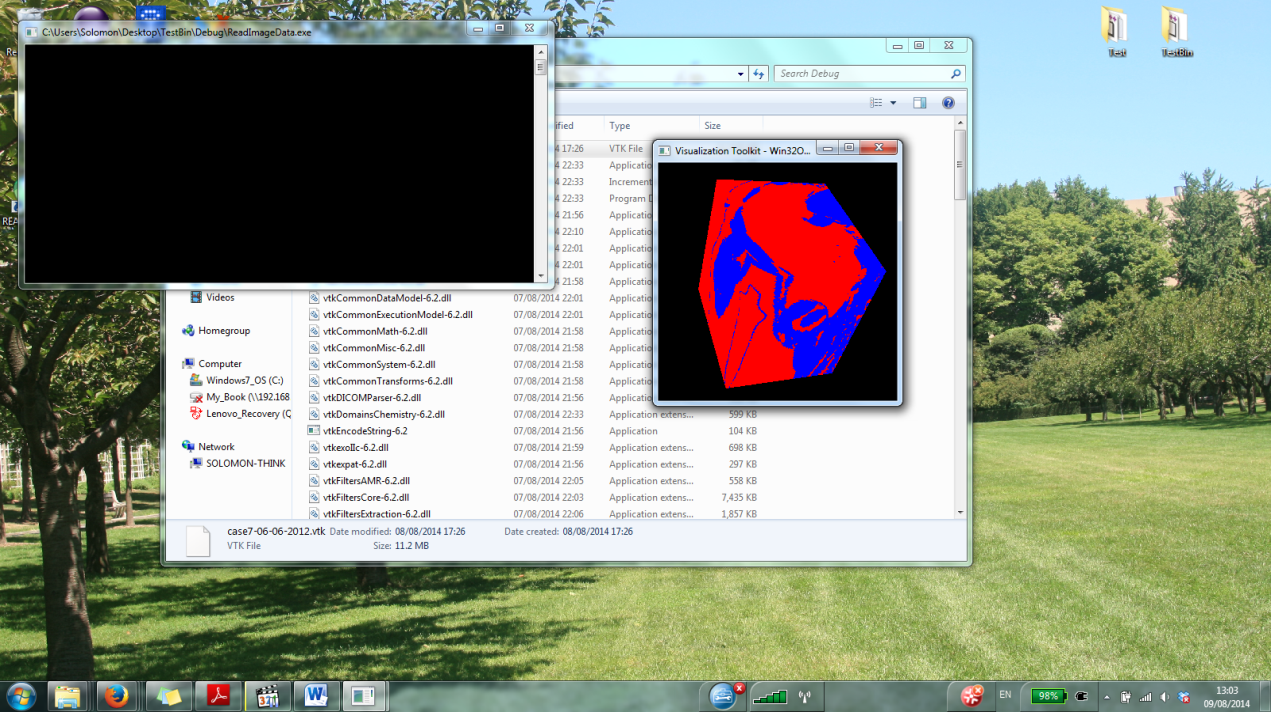
הרצה ידנית מCMD מצליחה, אך הדוגמא שבה אנו משתמשים מיושנת, בVTK 6 יש שינוי כך ש<vtkImageActor>.setInput אינו קיים יותר והשימוש בגרסא החדשה, setInputData מתחילה לרוץ אך מראה חלון שחור ללא הCT. <אחיה הבטיח לבדוק את העניין ביום ראשון ה10 ביולי.



הוסף הפרוייקט הנ"ל בקובץ ExampleSlicesVis .

**09.08.2014**

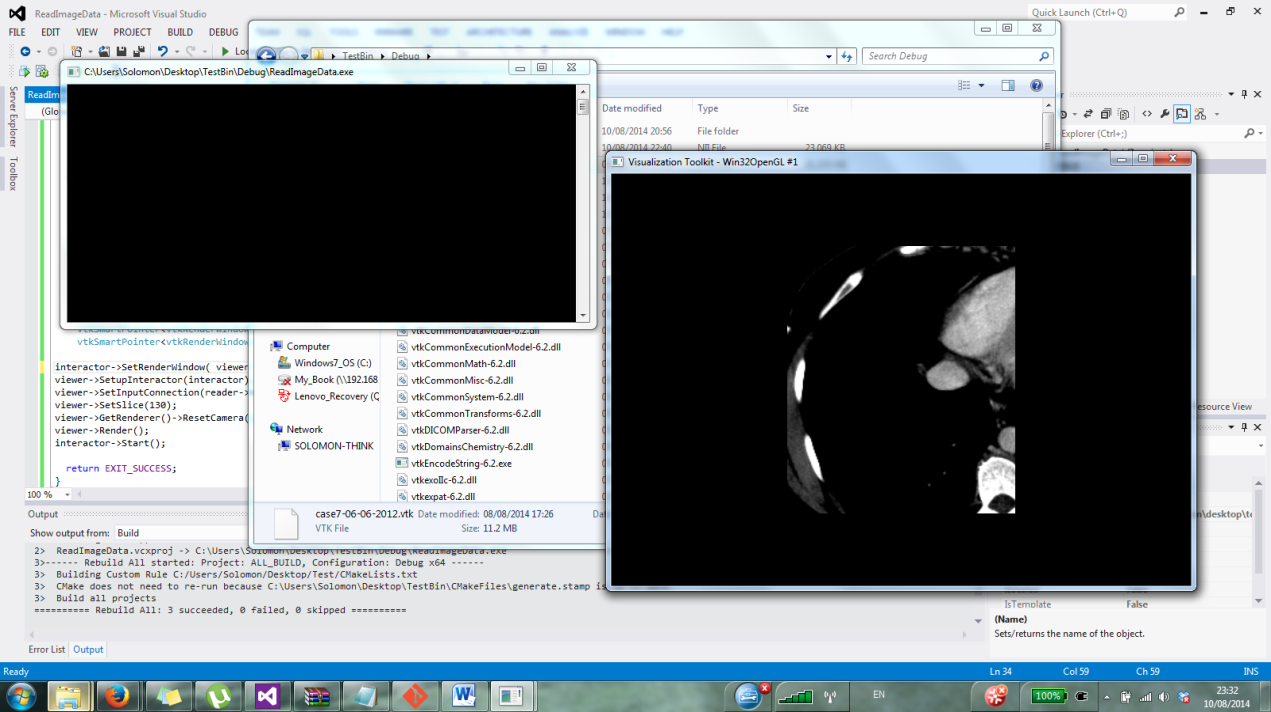
הבעייה תוקנה ע"י החלפת הקוד לגרסא מחודשת.



הבעייה כעת היא צבע אדום כחול משום מה, וכפי שניתן לראות בתמונה הVIEWS נמצאים על קובייה תלת מיימדית כך שהאינטרקציה עם החולון היא הזזה של הקובייה ולא מעבר בין SLICES.

**10.08.2014**

הבעייה תוקנה, כעת הUI וכן התמונה היא כSLICE של CT אמיתי. הקריאה לפי STRUCTURED POINTS וההצגה ע"י IMAGEVIEWER2. להלן התמונה:



נותר להוסיף custom interactor, דוגמא לכך נמצאת בEXAMPLE של VTK הנקרא ReadDICOMSeries.

**12.08.14**

הוסף CUSTOM INTERACTOR עם יכולת דפדוף SLICES בעזרת מקלדת ועכבר, שינוי בהירות(CONTRAST) ושינוי אורינטציה.

**13.08.14**

ניסיון לחבר בין הLEAP לבין פרוייקט הVTK.

נתקלנו בבעיה של VTK לרנדר את החלון של הImageViewer מהEventListener של הLEAP.

מסתבר שהLEAP מקבל את הEVENTS שלו בThread נפרד וVTK לא מאפשר UI ורינדור משני

Threads שונים. להלן התמונה של השגיאה שקוראת:

**15.08.14**

הגענו למסקנה שהדרך היחידה לשתף את הLEAP. הינו דרך שינוי מספר הSLICE בINTERACTOR מהLEAP EVENT והוספת טיימר שכל 15ms מרנדר את החלון עם הסלייס הנתון.

גישה זאת פעלה הייטב, כעת ניתן להעביר SLICES בעזרת קירוב והרחקת האצבע מהמסך ע"י הLEAP.

הפסקנו להשתמש בCMake בכדי לקמפל את הפרוייקט. לצורך כך ולצורך סידור נוח של הספריות

חשוב לזכור כי יש להוסיך את שתי השורות הבאות לפני INCLUDE של ספריות הVTK:

#define vtkRenderingCore\_AUTOINIT 4(vtkInteractionStyle,vtkRenderingFreeType,vtkRenderingFreeTypeOpenGL,vtkRenderingOpenGL)

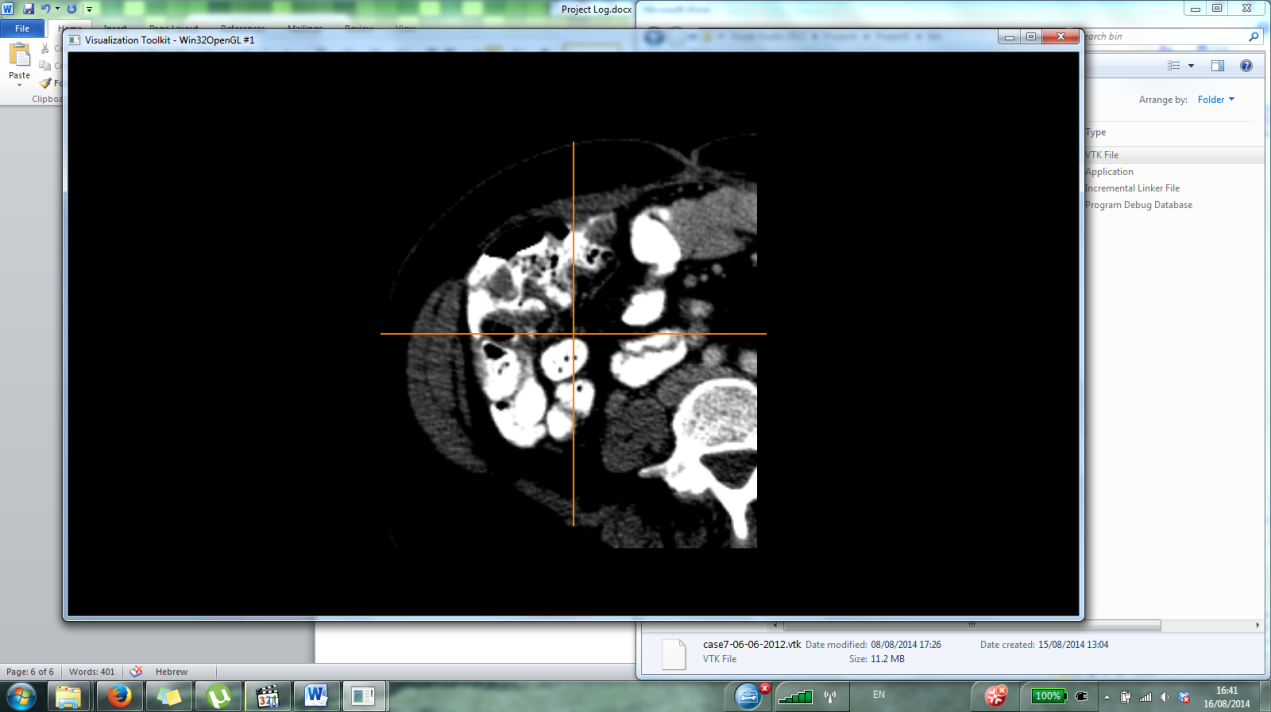
#define vtkRenderingVolume\_AUTOINIT 1(vtkRenderingVolumeOpenGL)

וידאו של אופן פעולת התוכנית יתווסף במהרה.

נותר להוסיף '+' על המסך.

**16.08.14**

נוסף '+' כתום על המסך ע"י הוספת Renderer לחלון של ה ImageViewer.



**2.09.14**

הוספנו אפשרות להזיז את ה'+' על המסך, וכן הצבע שלו משתנה מכחול לאדום בזמן לחיצה על CTRL, מקש הסימון שלנו, אם כי ה'+' יוצא מהגבולות ולא נמצא בקואורדינטות התמונה לכן השימוש בו קצת קשה ופחות אינטואיטיבי. אנו נתקלנו בבעיה רצינית בזמן ציור על המסך.

נתקלנו בבעיה כאשר ניסינו לסמן על המסך סגמנטציה. הבעיה הראשונה הייתה חוסר היכולת להוסיף MASK עליו נוכל לצייר. הוספנו רנדרר נוסף לחלון אך כל מה שראינו היה ריבוע תלת מימדי

מעל לתמונת הCT שלנו.

פתרנו בעיה זו ע"י הוספת הMASK כACTOR לאותו הרנדרר של הCT.

הגענו למסקנה שהדרך הטובה ביותר לצייר על הMASK היא להוסיף לו LOOKUPTABLE שממפה נק.

עם סקלר 0 לחסר צבע ו1 לאדום ואז לשנות את הסקלרים בזמן הציור.

נתקלנו בבעיה כאשר עדכנו את הסקלרים אך התמונה לא השתנתה. פתרון הבעיה היה לקרוא ל

<scalar\_array\_of\_the\_structure>.Modified(). הפונקציה גרמה לפיפלין לעדכן את המידע שלו.

**5.09.14**

ניסינו לתקן את הבעיה בה עצרנו בפעם האחרונה. הצלחנו לגרום לMASK להשתנות לאדום רק ב20000 הנק. הראשונות ע"י לולאת פור ורק על הסלייס הראשון. כל ניסיון לצייר נק. בודדת במקום כלשהו בנפח התלת מימדי היה כישלון וכן ה'+' היה מחוץ למערכת הקואורדינטות לכן סימון לא היה יכול להיעשות עפ"י המידע של הקרוס.

בעיה נוספת הייתה שמהירות התגובה של תזוזת ה'+' ירדה מ"חלקה" בזמן תזוזה רגילה ומעבר בין סלייסים ל"לא סבירה" בזמן סימון, לחיצה על CTRL".

מכיוון שלא הצלחנו לפתור את בעיית הסימון נקודות, עברנו לחשוב על בעיית הPerformance drop.

הגענו למסקנה שהפייפלין מרנדר בכל רגע נתון לא רק את הסליסים הנראים אלו שהם בפרסטרום אלא את כל 131 הסלייסים יחדיו. לכן זמן התגובה היה גרוע.

החלטנו להשתמש בVOIExtractor יצור של VTK שנותן אפשרות לחתוך נפח מתוך נפח אחר על-מנת להוציא סלייס יחיד ולהראות אותו על המסך. הצלחנו להוציא סלייס יחיד אבל הVIEWER התקשה לשנות את הסלייס ברגע לחיצת SHIFT וכן זמן הEXTRACTION היה ארוך מדי ועדיין הPERFORMANCE היה גרוע.

**22.09.14**

עזבנו את הרעיון של הוצאת סלייס יחיד ע"י VOI והבנו שאנו עושים משהו לא נכון והVIEWER צריך להחליט בעצמו לרנדר רק את הפרסטרום. מצאנו בVTKUSERS מישהו שעשה ב2012 GUI דומה של סימון סגמנטציה על המסך, למרות שהשימוש היה שם בQT והפרוייקט התבסס על VTK5.8 החלטנו לכתוב ליוצרים מייל ולבקש עזרה.

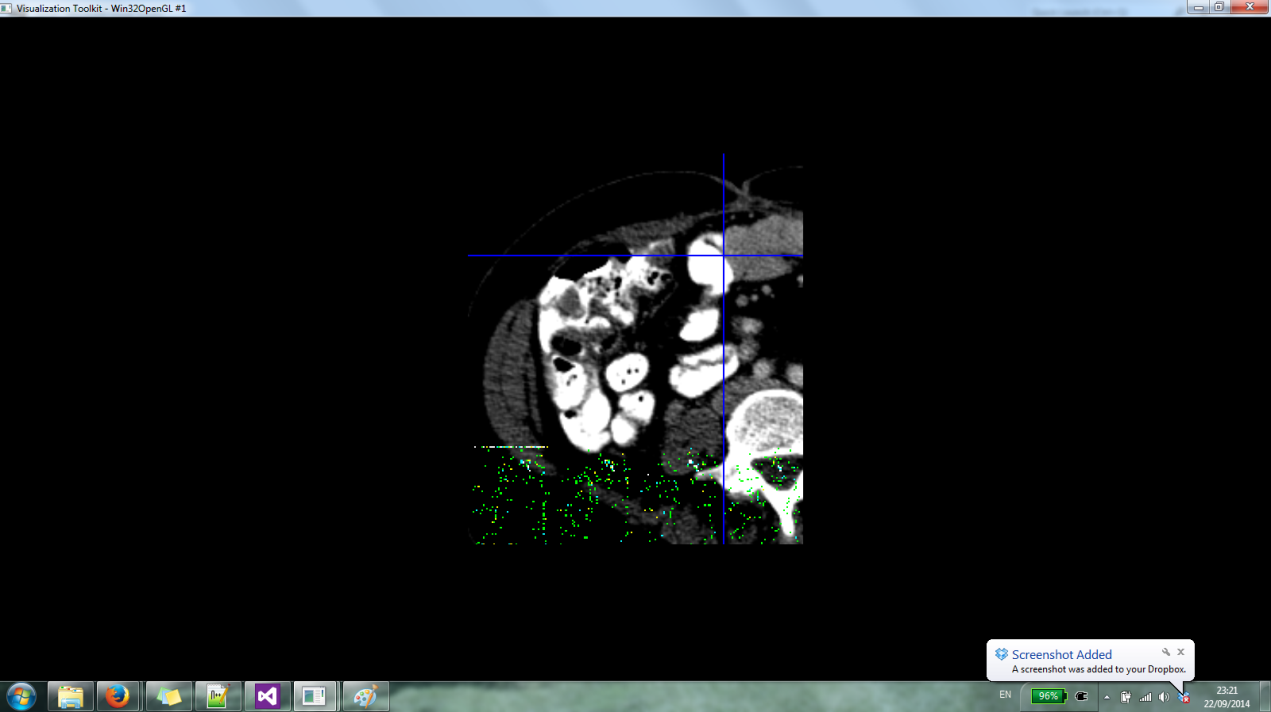
למזלנו היוצר שלח לנו כמה פיסות קוד מאד שימושיות.

מתוך הקוד החלטנו לעבור למאפר מסוג vtkImageMapToColors במקום הvtkDataSetMapper.

ובאקטור לעבור לvtkImageActor במקום vtkActor.

כמו כן העברנו את ה'+' להיות גם הוא actor באותו הרנדרר עם הCT מה שהצריך הרבה מעברי קואורדינטות כדי להביא את שתי היצורים לאותו המימד.

לאחר הרבה תיקונים ושיפצורים בקוד הצלחנו להגיע לתוצאה מסויימת. הPerformance נהייה טוב מאד ללא ירידת זמן תגובה נראה. והצלחנו לצייר על המסך אבל עדיין הסימון לא נראה טוב. ראינו שהוא לא עבר את המאפינג של הLOOKUPTABLE, וכן הוא היה קטן ולא השתרעה על כל המסך ולא היה מסונכרן עם ה'+'. להלן תמונה:



**23.09.14**

שמנו לב בפעם שעברה שלא יכולנו להוציא את הLOOKUPTABLE מהמאפר, תמיד קיבלנו NULL וכן נזכרנו שאחד השיפצורים שלנו היה להחליף את השמת המאפר מהפקודה:

<vtkImageActor\_of\_selection>.setInput(<mapper>)

לשורה שלוקחת בחשבון את השינוי מVTK 5 ל6:

<vtkImageActor\_of\_selection>.getMapper()->SetInputConnection(<mapper>->GetOutputPort())

לכן עשינו את השינוי המתאים גם בהוצאת המאפר מהאקטור ופתאום הכל התחיל לעבוד.

הסימון האדום הופיע בדיוק בנקודות עליהן היה ה'+'. להלן סרטון הדגמא:

<https://www.youtube.com/watch?v=3zwp70_u-jM>

יותר מאוחר הוספנו גם אפשרות למחוק את הסימון ע"י לחיצה על ALT.

**נרשמנו לIntel RealSense Challange.**

**הזכרנו לדורון להזמין לנו LEAP.**

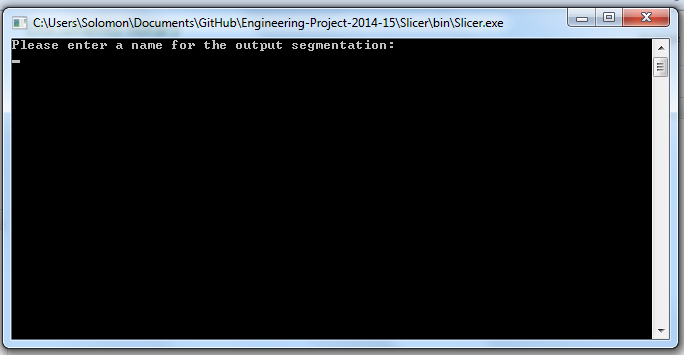
**24.09.14**

נוספה היכולת לשנות את עובי הסימון ע"י לחיצה על מקשים 1-להקטנה ו2- להגדלה.

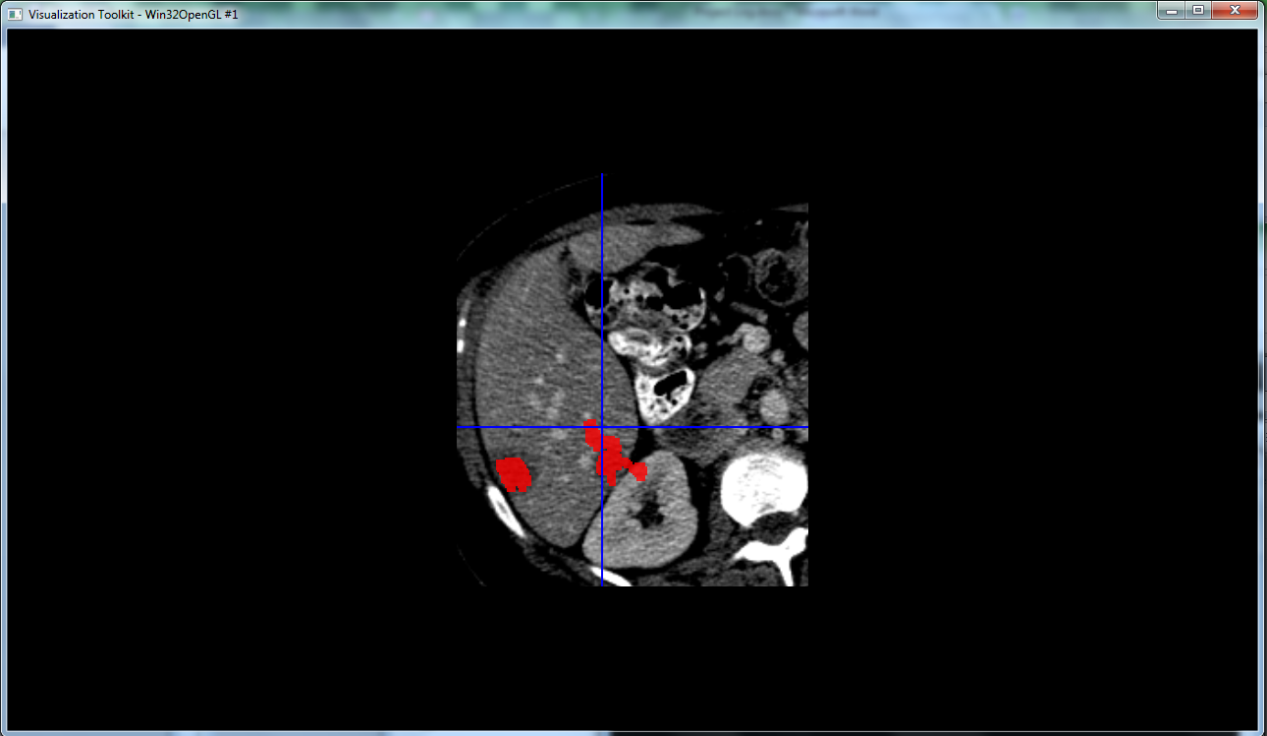
הקוד נוקה.

**25.09.14**

נוספה היכולת לשמור את סימון הסגמנטציה לקובץ. כרגע מזמן התחלת התוכנית נותנים בCMD את שם הקובץ לOUTPUT וכשרוצים לשמור לוחצים על s . תמונה:



והסימון נראה כך:

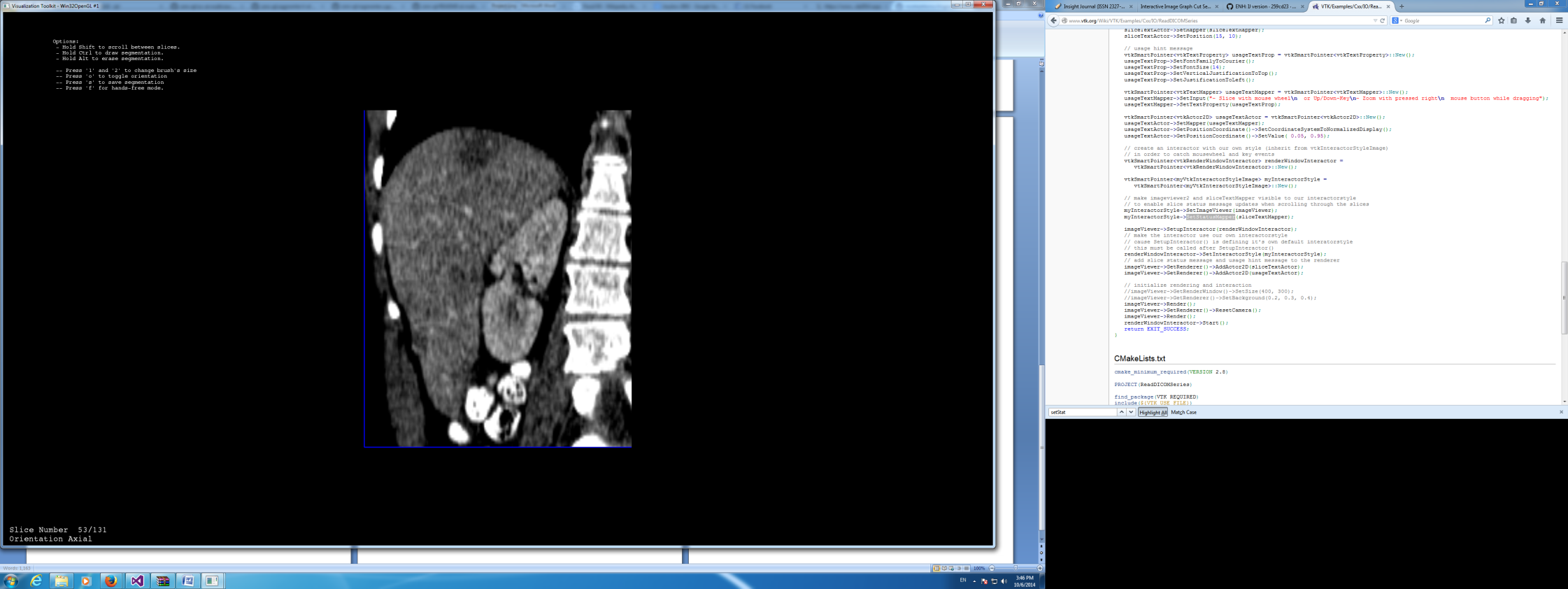


בהמשך נעבור לGUI נוח יותר כנראה עם Qt, אך בשביל זה צריך לקמפל מחדש את VTK וזה מפחיד מדי.

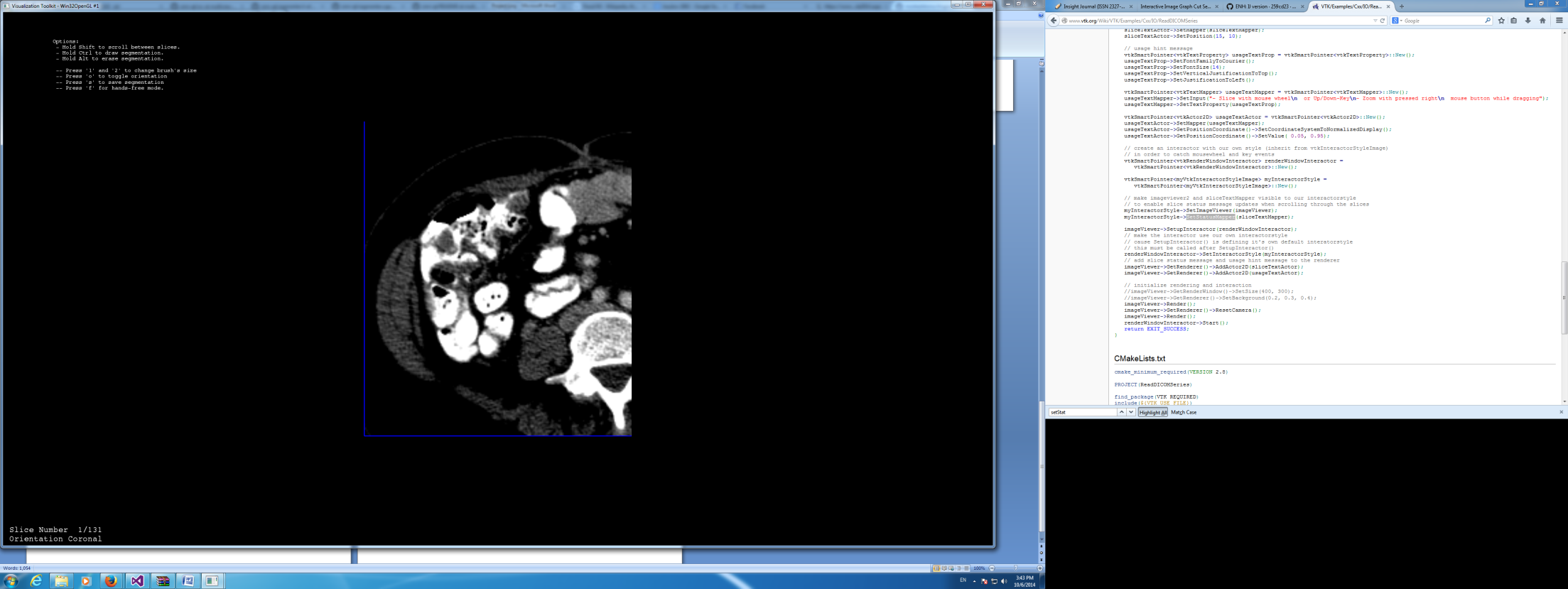
נוספה אפשרות Hands free ע"י לחיצה על מקש 'f' . פתיחת האגודל לסימון ופתיחת האצבע האמצעית למעבר בין סלייסים.

**6.10.14**

תיקנו אורינטציות (סימון ומעבר). (בתמונה הבאה יש דוגמא לאוריינטציה Sagittal (YZ)).



הוספנו מדיע ויזואלי על החלון לגבי אפשרויות, מספר הסלייס והאורינטציה הנוכחית. (רואים בתמונה מצד שמאל למעלה ולמטה).



כרגע אנחנו מתלבטים בין שני אלגוריתמים בשביל לבצע סגמנטציה: graph cut OR flood fill. אמורים להיפגש עם רפי מחר בשביל לדסקס על זה.

**20.10.14**

התחלנו במימוש של graph cuts. כרגע ממשים פונקציות בסיסיות ומעבירים לעבודה עם vtkstructurePoints.

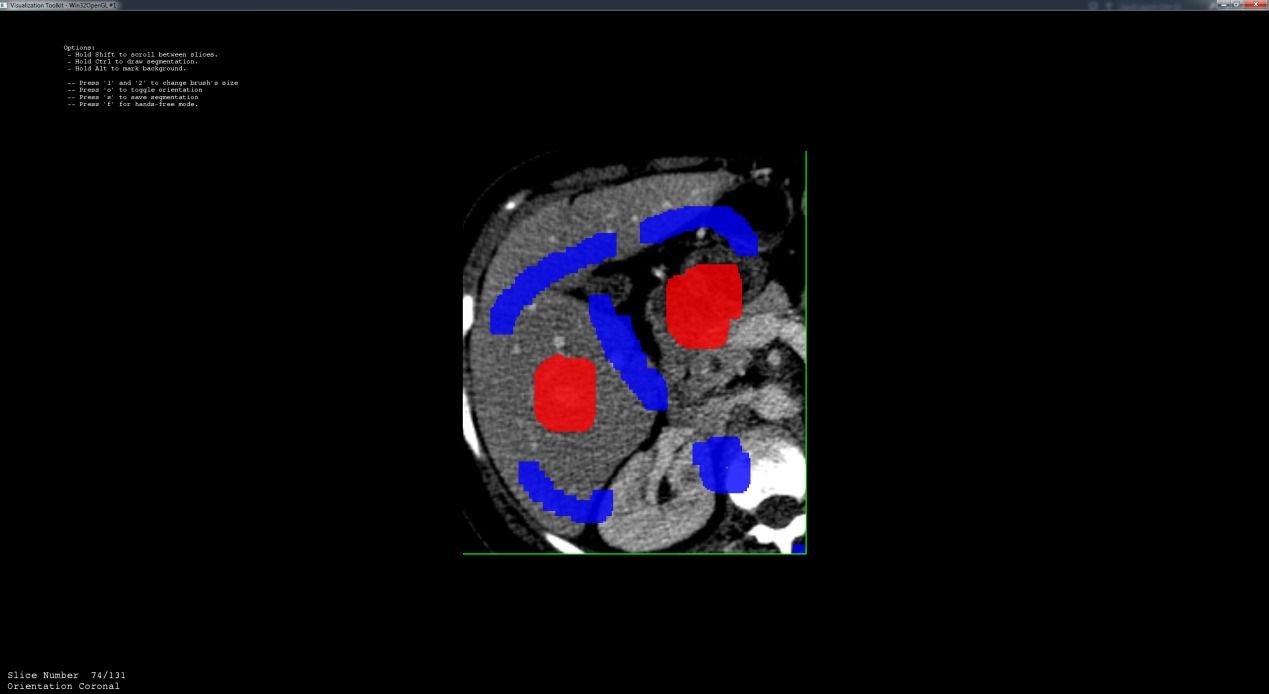
לא מתקמפל כרגע.

החלפנו את האפשרות למחיקה בסימון "רקע" כדי שיהיו source and target.

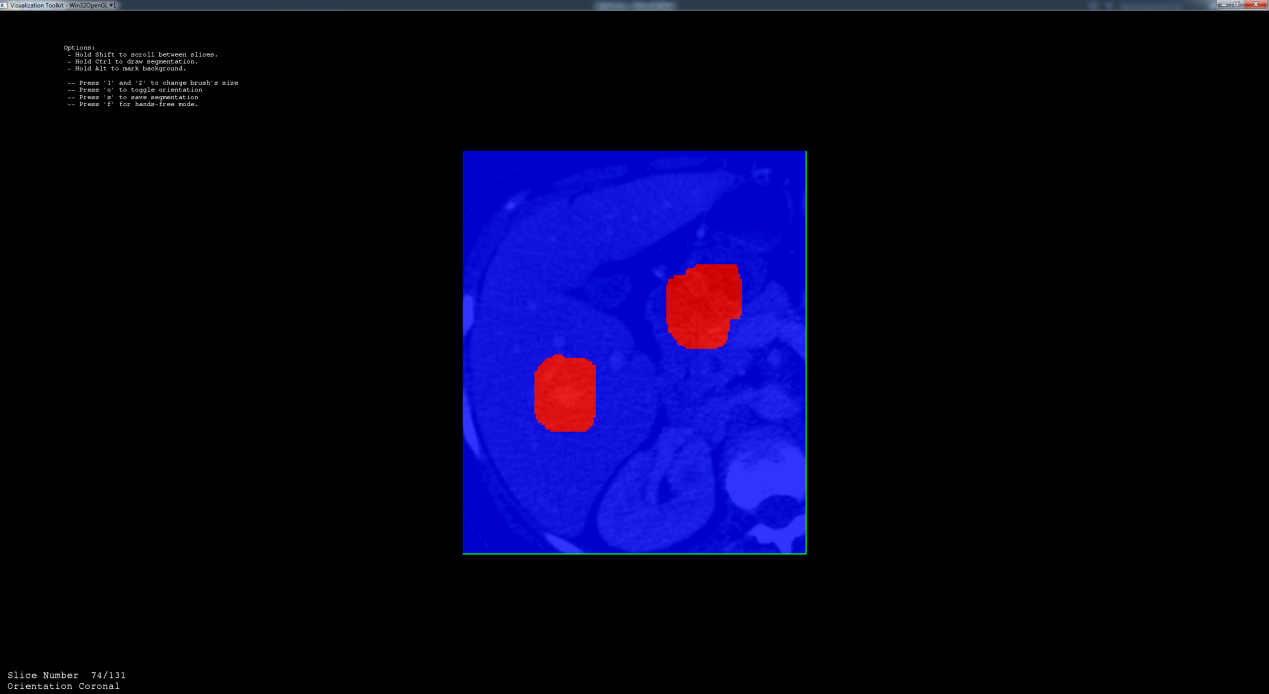
**26.10.14**

GraphCuts מבצע את מה שהוא אמור לעשות. הייתה בעיה של זמן ריצה, אבל היא נפתרה אחרי שסידרנו את המצב של ה-release עבור הVTK. עכשיו זה עובד בשניות בודדות.

כרגע האלגוריתם מתייחס רק לשלב הראשון של האילוצים: ניתן לסמן לו מה גידול ומה לא גידול, והוא פשוט בוחר רק את מה שהוא לא גידול. לשם הבהירות אנחנו גם מסמנים את כל השאר כלא גידול. (אדום וכחול בהתאמה, דוגמא לעיל).



בתמונה שלמעלה אנחנו רואים סימון פשוט. בתמונה שלמטה אנחנו רואים את תוצר ריצת האלגוריתם (שגם הוא פועל ברקע על thread נפרד). זמן הריצה: 8 שניות, וזה כולל את היצירה של הגרף עצמו (פעולה שלוקחת יחסית יותר זמן).



אחרי זה התחלנו לעבוד על להוציא התפלגות מתוך ה-GT שקיבלנו מרפי. יש מספיק קייסים. אנחנו בשלבי סיום של סקריפט שעובר על כל הקבצים בתיקייה מסויימת, ומוציא עבור כל אחד מהם את ה-mu ואת ה-sigma בהנחה של התפלגות נורמלית. בשלב הבא נרצה להשתמש בידע שנקבל מזה בשביל להוסיף עוד אילוצים על הצלעות ך-source ול-target באלגוריתם. (T-edges).