



Augmented Indoor Navigation

שמות המגישים : ארקדי גורביץ', שחר זיוון

שם המנחה : ירון חונן

תאריך הגשה : 26.04.2020

תוכן עניינים

מבוא ותקציר	4
תיאור המערכת	5
תיאור הביצוע	7
תוצאות	11
מסקנות	12
המלצות	13
מקורות ספרותיים	14

רשימת איורים

1. [תיאור המערכת ניסיון ראשון](#)
2. [תיאור המערכת ניסיון שני](#)
3. [מכשירי הביקון](#)
4. [אפליקציה לדוגמא למדידת מרחק מביקונים](#)
5. [המרקרים שבהם השתמשנו](#)
6. [מפת המשרד ששימש לבדיקות](#)

מבוא ותקציר

מטרת הפרויקט היא לאפשר ניווט בתוך בניין באמצעות בלוטוס ומשקפי HoloLens . המשתמש מתחיל מנקודת התחלה כלשהי בתוך המבנה, בוחר את נקודת היעד ע"י בחירת יעד מתוך תפריט ומקבל ניווט אינטראקטיבי באמצעות המשקפיים.

ישנם פתרונות אשר מתבססים על מד התאוצה והמצפן של הפלאפון. הבעיה עם פתרונות כאלה שהם תלויים בסוג הטלפון. טלפונים עם סנסורים לא טובים מספיק לא יספקו דיוק מספיק טוב. לכן צריך פתרון שהוא רובסטי לסוג הטלפון. בפתרונות נוספים אשר מתבססים על Bluetooth היה צורך להשתמש במספר גדול של ביקונים (48) ולהיעזר בכלים נוספים לתיקון שגיאות. כמו כן יש פתרונות המתבססים על מציאות רבודה אך ללא חיפושים יעילים בגרף ולכן מאפשר ניווט בסיסי בהחלטה.

הפתרון שלנו בסופו של דבר הלך לכיוון אחר. הצגנו פתרון באמצעות Vuforia המתבסס על עיבוד תמונה גרידא. בניסיון הראשון, השתמשנו ב-3 מכשירי בלוטוס אך לא הצלחנו לקבל דיוק טוב מספיק ולכן עזבנו את הכיוון הזה. הפתרון המוצע הצליח להביא דיוק רב ומציאת מסלול מהיר ביותר.

המסקנות שלנו הן שכנראה היינו צריכים להשתמש בכמות גדולה יותר של מכשירי בלוטוס וגם המכשירים עצמם צריכים לשדר בעוצמה גדולה יותר. פתרון בעזרת Vuforia נותן תוצאות טובות אך מצריך יותר עבודה.

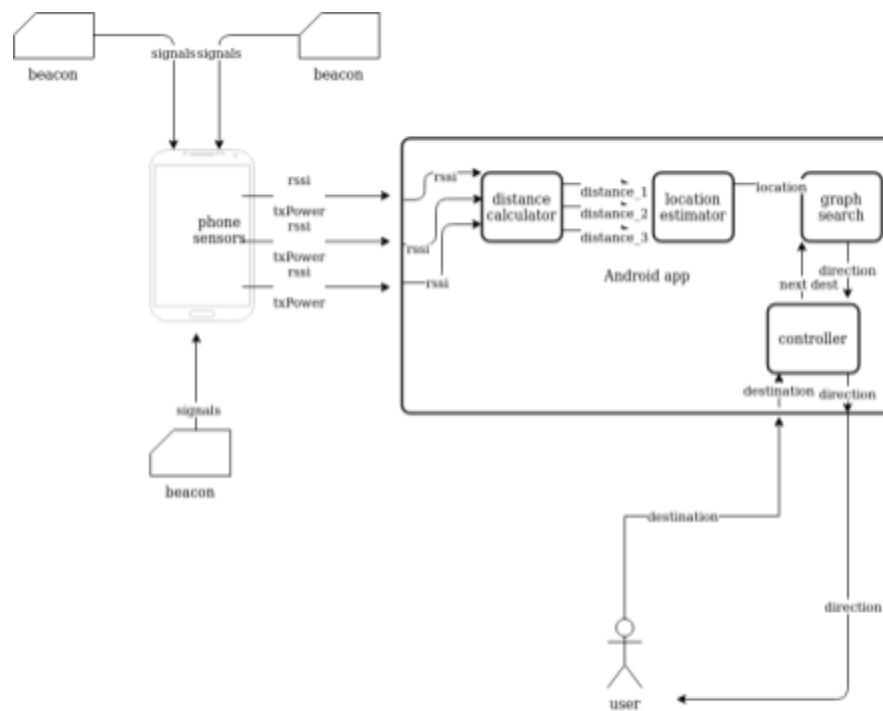
ההמלצות שלנו הן לנסות לשפר את מה שעשינו באמצעות מכונות לומדות שיקלו את העבודה במעבר ממבנה אחד לשני.

תיאור המערכת

ניסיון ראשון:

ציוד נדרש:

1. 3 ביקונים
2. משקפי HoloLens
3. טלפון אנדרואידי



איור 1 : תיאור מערכת ניסיון ראשון

השתמשנו ב 4 מודולים :

1. distance calculator - מקבל מדידות rssi של כל אחד מהביקונים ומוציא את המרחק מכל אחד מהם
2. location estimator - מחשב את המיקום המדויק ע"י triangulation
3. graph search - מבצע חיפוש בגרף של הנקודה הבאה במסלול. מקבל בהתחלה את היעד ולאחר מכן את המיקום ומחזיר את כיוון ההליכה . החיפוש נעשה באלגוריתם A*. בכל שלב חיפשנו את המסלול הטוב ביותר והחזרנו את הנקודה הבאה במסלול

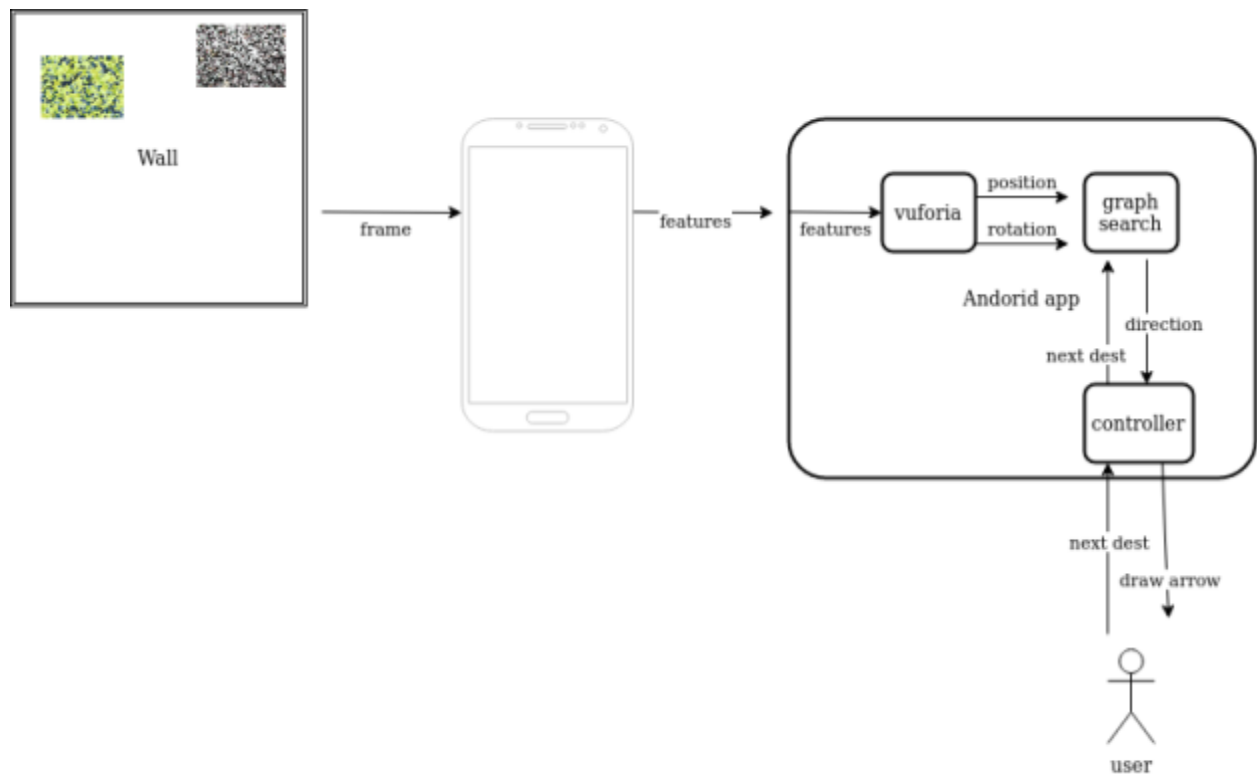
4. controller - מקבל מהמשתמש את היעד, ושולח אותו לחיפוש בגרף, לאחר מכן מקבל מהגרף את כיוון ההליכה ומחזיר אותו למשתמש כל המודולים נמצאים באפליקציית האנדרואיד שבנינו.

ניסיון שני:

ציוד נדרש :

1. טלפון אנדרואידי
2. משקפי HoloLens
3. תמונות/פוסטרים שימשו כמרקרים בבניין

תיאור המערכת :



איור 2 : תיאור מערכת ניסיון שני

למעשה המערכת מורכבת מ 3 מודולים.

1. ה Vuforia - שתפקידה לקבל את הפיצ'רים מהטלפונים ולפי מיקומים שהוגדרו מראש תדע להחזיר את המיקום והאוריינטציה של המשתמש.
2. ה graph search - בגרף משתמשים על מנת למצוא את המסלול הקצר ביותר ע"י אלגוריתם *A והוא למעשה מחזיר את הנקודה הבאה במסלול
3. ה controller - הוא המנהל של הכל. מקבל מהמשתמש את היעד ולאחר מכן, לאורך המסלול מסמן למשתמש לאן צריך להמשיך.

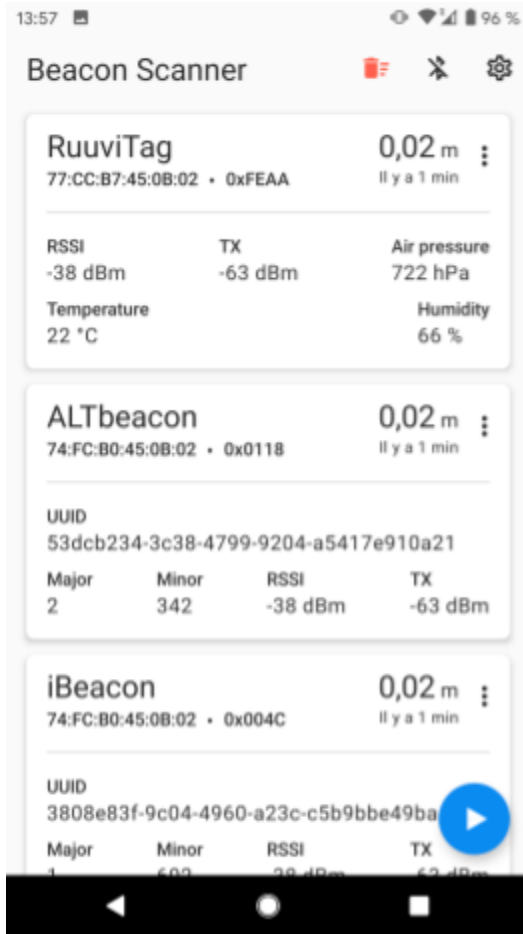
תיאור הביצוע

בניסיון הראשון עבדנו עם 3 מכשירי בלוטוס, מסוג iBeacon, ששידרו אותות BLE. BLE היא סוג של תקשורת אלחוטית המיועדת במיוחד לתקשורת לטווח קצר. BLE דומה מאוד ל-Wi-Fi במובן זה שהוא מאפשר למכשירים לתקשר זה עם זה. הטלפון קיבל אותות מכל אחד מהמכשירים וחישב מרחק אבסולוטי מכל אחד מהם. מהאותות שהתקבלו חילצנו שני פרמטרים חשובים:

1. rssi - מדד לעוצמת האות של הביקון כפי שנמדד במכשיר המדידה (במקרה שלנו זה הטלפון האנדרואידי)
2. txpower - שמציין מה ערך ה rssi במרחק של מטר אחד ממנו

באמצעות 2 הפרמטרים האלה חישבנו את המרחק של הביקון מאיתנו וזה ע"י מיצוע ה rssi על גבי הזמן בכדי "להחליק" את האות.

בנוסף בהגדרות הסריקה (ScanSettings) השתמשנו באפשרות של "SCAN_MODE_LOW_LATENCY" על מנת לקבל את האות בצורה בתדירות גבוהה. שיחקנו הרבה מאוד עם ההגדרות אך בכל הניסיונות שלנו לא הצלחנו לקבל דיוקים מספיק טובים, עדכון המרחק היה איטי מדי ולא מדויק, גם כשניסינו בסביבות סטריליות לגמרי, כלומר ללא מכשירים נוספים שמשדרים בלוטוס. בנינו אפליקצית אנדרואיד על מנת שנוכל לבדוק בצורה נוחה את חישובי המרחק ולאחר שלא הצלחנו, השתמשנו באפליקציה [\[1\]](#) שמצאנו בgithub, למשל אחת בשם android-beacon-scanner שראינו שעובדת לא רע על מכשירים אחרים ששידרו אותות בלוטוס אך לא עבדה טוב עם המכשירים שלנו. עשינו המון בדיקות, בתוך מבנה, מחוץ למבנה, ניסינו בסביבה קרובה ובסביבה רחוקה אך אף פתרון לא עבד. בהנחה וחישוב המרחק ממכשיר בלוטוס היה עובד, היינו עושים חישוב טריאנגולציה על מנת לחשב את המיקום המדויק במרחב ואז מבצעים חיפוש בגרף אך החלטנו ללכת לכיוון אחר.



איור 4 : אפליקציה לדוגמא



איור 3 : מכשירי הביקון

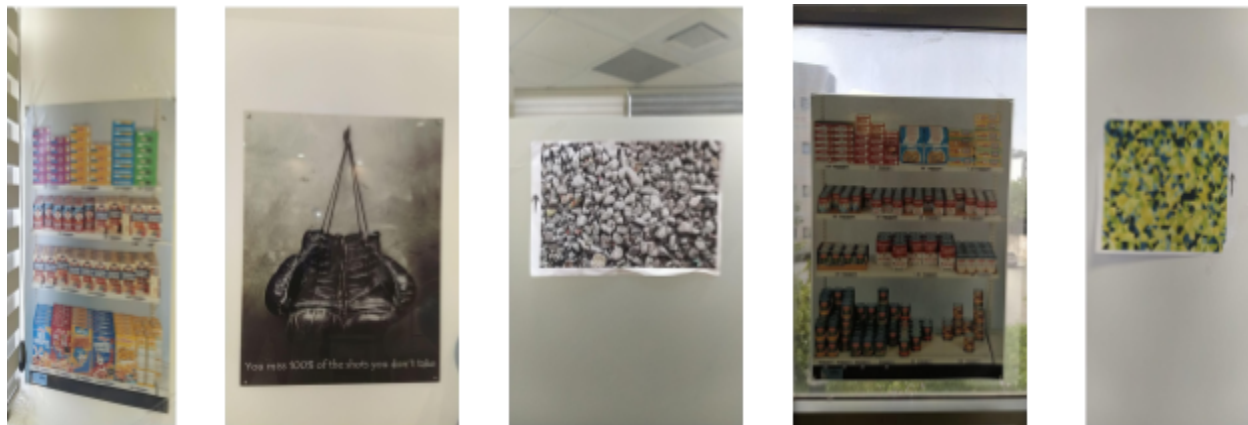
בניסיון השני עבדנו עם Vuforia. יעדי מנוע Vuforia של חברת ptc מאפשרים למפתחים להשתמש בחלל שלם, בין אם מדובר בקומת מפעל או בחנות קמעונאית, כיעד ריאליטי מוגדל. בניסיון הזה תלינו פוסטרים שישמשו את ה Vuforia כמרקרים. הכנסנו אותם לתוך מבנה הנתונים של ה Vuforia ובנוסף הם היו שמורים בתוך הסצנה של Unity. בתוך מבנה הנתונים של Vuforia הגדרנו את גודל התמונות ובתוך ה Unity הגדרנו את המיקום שלהם בתלת מימד ואת וקטור הנורמל והצלחנו לקבל מיקום דיי מדויק ביחס למרקרים. בהינתן שידענו איפה תלינו את המרקרים בעולם האמיתי, ידענו לחשב את המיקום המדויק בעולם האמיתי. הפתרון עובד בצורה טובה מאוד.

האלגוריתם:

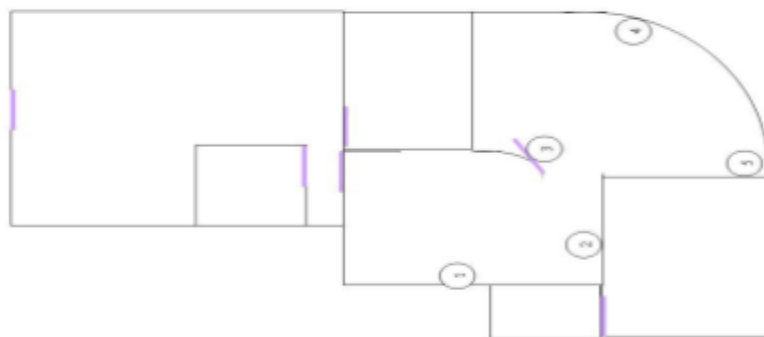
1. קבל מ Vuforia מיקום ואוריינטציה של המשתמש ביחס למרקר שראית
2. חשב מיקום בעולם האמיתי ע"י שימוש במיקום האבסולוטי של המרקר בעולם האמיתי
3. בצע חיפוש *A בגרף לנקודת היעד
4. החזר את הצומת הקרוב ביותר
5. צייר חץ לצומת הזה

תיאור הביצוע של המערכת :

תחילה יוצרים גרף של המבנה. הגרף כולל את הנקודות המעניינות לאורך המסלול, עם מיקום מדויק שלהם במערכת הצירים שגם אותה קובעים מראש בצורה נוחה למתכנת. משתמשים בתמונות או פוסטרים בתור מרקרים עבור ה Vuforia. את המרקרים מכניסים למבנה הנתונים של ה Vuforia ומגדירים את המיקום שלהם (מגדירים את ה x, y, z ואת וקטור הנורמל) ובנוסף מכניסים אותם לתוך הסצנה ב Unity. כאשר המשתמש מתחיל בניווט, הוא בוחר יעד מתוך רשימה וברגע הראשון שמופיע מרקר בשדה הראייה של המשקפיים מתבצע חיפוש *A בגרף ומוחזר הצומת הקרוב ביותר. לאחר מכן המערכת משתמש במצפן של הטלפון כדי להמשיך במצב "עקיבה" ואז למעשה ניתן להתקדם גם כאשר אין מרקר בשדה הראייה של המשתמש. המערכת מציירת למשתמש חץ שמסמן לאן הוא אמור להמשיך. בנוסף המערכת מציירת בתוך מפה את מיקום המשתמש. התהליך נמשך עד שהמשתמש מגיע ליעד.



איור 5 : המרקריסופוסטרים שהשתמשנו בבדיקות



איור 6 : מפת המשרד

תוצאות

תוצאות בניסיון הראשון. בטבלה נציג מדידות כאשר השתמשנו ב 3 ביקונים . התוצאות מתייחסות למדידה אל מול אחד הביקונים. התוצאות בכל הביקונים היו לא טובות ולכן אין התייחסות למספר הסידורי של הביקון הנמדד. ניתן לראות שאין מגמה ברורה לתוצאות. תחילה היינו קרובים והתוצאות היו קרובות ואז התרחקנו התקרבו והתוצאות לא היו קשורות למציאות. ההרגשה הייתה שהאות היה חלש מדי.

מיקום אמיתי ביחס לביקון	מיקום שקיבלנו בתוכנה
0.5m	0.8m
1.5m	2.7m
3m	4m
2m	3.8m
1m	2m
0.5m	1.5m

תוצאות בניסיון השני. ביצענו בדיוק את אותו הטסט, על מנת שנוכל להשוות תוצאות, אל מול אחד המרקרים. כאשר התקרבו והתרחקנו ממנו. התוצאות היו ממש מרשימות. ניתן לראות שכאשר התקרבו והתרחקנו, היה זיהוי מדויק של המיקום.

מיקום אמיתי ביחס למרקר	מיקום שקיבלנו בתוכנה
0.5m	0.56m
1.5m	1.4m
3m	3.07m
2m	1.95m
1m	1m
0.5m	0.5m

מסקנות

בניסוי הראשון הגענו למסקנה שהבעיה העיקרית הייתה עם הביקונים. האות של הביקונים היה חלש מדיי ולכן לא התאפשר לנו לעשות מדידות טובות. אפשרות אחת לפתרון היא לקנות ביקונים עם אות חזק\יציב יותר. אפשרות שניה היא להגדיל את כמות הביקונים ובכך לפצות על הבעייתיות של חוזק\יציבות האות. בכל מקרה פתרון כזה יכול לעבוד במקרה הכללי.

המסקנות לגבי הניסוי השני הן שהשימוש ב Vuforia יכול להיות פתרון מקצה לקצה לבעיית הניווט אשר משיג תוצאות טובות ללא תלות במכשירים אלקטרוניים נוספים.

גם בפתרון הראשון, ניתן להיעזר ב Vuforia על מנת לתקן טעויות חישוב מצד אחד, ומצד שני לחסוך אולי להקטין את כמות המרקרים הדרושים.

המלצות

ישנם שיפורים רבים שניתן לעשות. קודם כל ניתן לעשות פלטפורמה ליצירת גרף המבנה בצורה דינמית. כלומר לייתר את הצורך להכין גרף מראש, אלא יצירת גרף אוטומטית בעזרת סריקה מהירה של האזור הנתון. שיפור נוסף שניתן לעשות הוא לשנות את הדרך שבה מקבלים מהמשתמש את היעד. ניתן לקבל את היעד ע"י פקודה קולית או ע"י שימוש בכלים של מציאות רבודה. שיפור\שינוי נוסף הוא הוספת תמיכה למכשירי IOS ולפתוח את השימוש לקהל יעד גדול יותר. שיפור נוסף הוא שימוש בכלים נוספים של הפלאפון כדי להגדיל את הדיוק. שיפור נוסף הוא שינוי אלגוריתם החיפוש בגרף ע"מ לקבל תוצאות טובות יותר.

מקורות ספרותיים

1. האתר לאפליקציה שבה השתמשנו כבדיקה נוספת לבלוטוס :
<https://github.com/Bridouille/android-beacon-scanner>