

Re: ELEC491: Fingerprint

Burak Soner <BSONER@ku.edu.tr>

Sun 3/10/2024 9:29 PM

To:ERDEM ERGEN <EERGEN20@KU.EDU.TR>;ABDULKADIR BILGE <ABILGE20@KU.EDU.TR>;BORA CARPINLIOGLU <BCARPINLIOGLU21@KU.EDU.TR>

2 attachments (254 KB)

pg.zip; Screenshot from 10-03-2024 19:40:09.webm;

Erdem, Kadir, merhaba,

Elinize sağlık. Aşağıda yorumlarımı paylaşıyorum.

Otomatik güncelleme yapan arayüz: bunun için ben küçük bir şablon hazırladım size, pygame isminde bir python kütüphanesi var onunla böyle şeyleri yapmak çok kolay oluyor. Sizin paylaştığınız resmi kesip küçültüp onun üstünde ekran koruyucu gibi sekerek gezinen bir yeşil top koydum. Kodu ve kesilmiş resmi zip dosyası olarak ekledim. Ekteki video'daki gibi bir görüntü çıkartıyor. Siz bunu alıp üstünde değişiklikler yaparak arayüzü kurabilirsiniz (başka kütüphaneler de eminim vardır, size daha iyi bir opsiyon gibi geleni varsa onu da kullanabilirsiniz). Bu pygame'in kurulumunda [şöyle](#) bir hata yaşarsanız, ben Solution 2'yi yaptım vaktinde (bende miniconda üstünden bir python venv kuruluydu), çalışıyor, aklınızda bulunsun.

Deneyiniz: Burada tam anlamadığım bir şey oldu, deneyi şöyle yaptınız sanırım değil mi:

- Sarı noktaların olduğu yerlerde 3 AP'den gelen RSS değerlerini kaydettiniz (buna "database fazı" diyelim)
- Sonra sarı noktaların olmadığı yerlerde 3 AP'den gelen RSS değerlerine bakıp, sarı noktalardan hangisine en yakındasınız diye euclidean distance baktınız, ve "observed location" dediğiniz değerleri elde ettiniz (buna "test fazı" diyelim)

Böyleyse bu OK, fakat bu deneyin sonuçlarını analiz edebilmek için şu bilgilere de ihtiyacımız var:

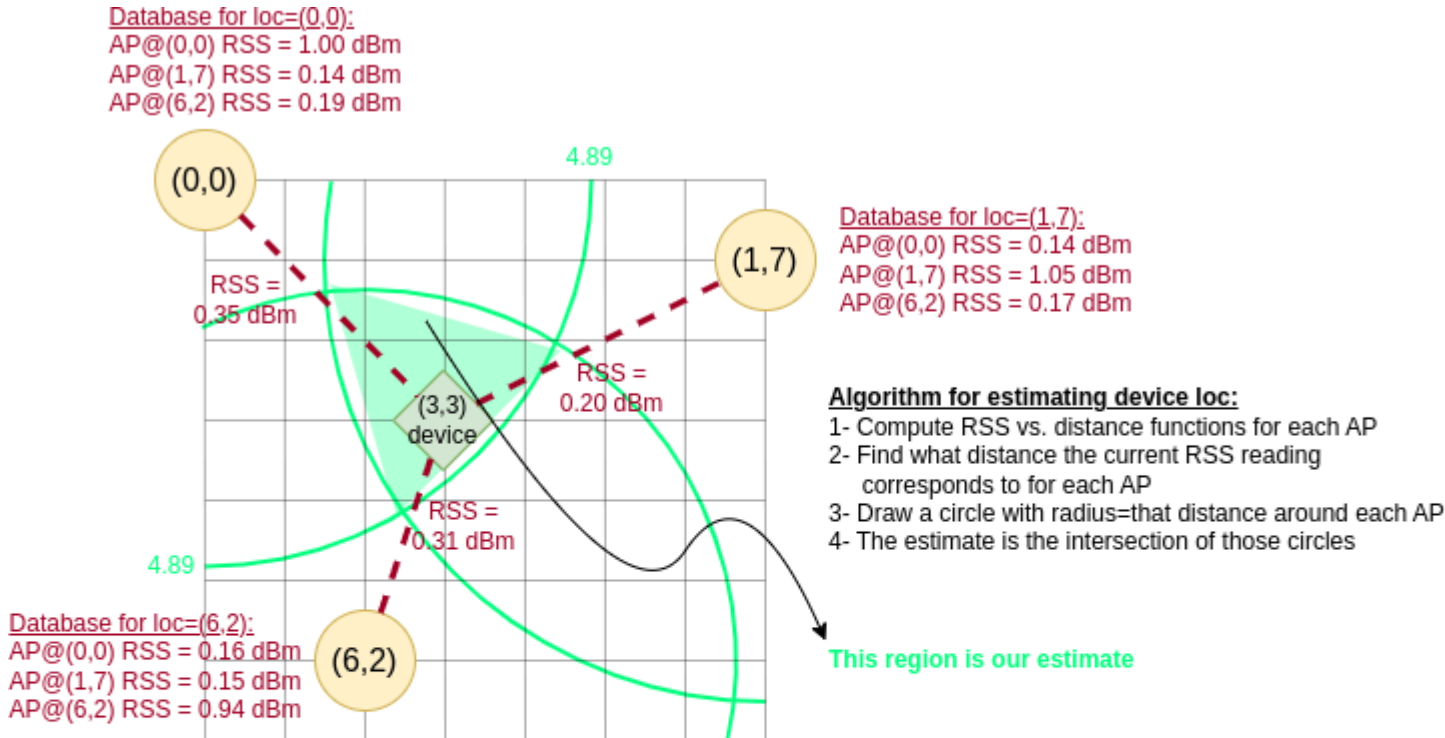
- Sarı noktaların hepsinin x,y şeklinde kordinatları. Toplam 14 tane var anladığım kadarıyla, bunları ya resmin üstüne çizerek ya da bir liste halinde paylaşabilirsiniz çok iyi olur. Resmin üstünde bir de neresi 0,0 onu da bilmiyoruz şu anda.
- Test fazında ölçüm yaptığınız yerlerdeki 3 AP'den gelen RSS değerleri (paylaştığınız tabloya 3 kolon daha ekleyip bunları da yazabilirsiniz, öyle daha rahat olur)

Bunları da paylaşabilirsiniz daha sağlıklı yorum yapabilirim.

Algoritmanız: Mevcutta anladığım kadarıyla siz bir küme üstünden "exact match" bakmışsınız, test fazında ölçüm yaptığınız lokasyonlardan her biri için algoritmanız en yakındaki sarı nokta nerdedir bunu hesaplıyor. **Yani mevcutta algoritmanın çıktısı hep bir sarı nokta lokasyonu, sarı noktalar dışında bir yerin lokasyonunu çıkartmıyor, bu doğru mu?**

Böyle formulate ettiğiniz zaman "ben bu kordinat düzleminde nerdeyim" probleminden ziyade biraz daha "ben bu kordinat düzlemindeki bilinen noktalardan hangisine en yakındayım" problemine bakmış oluyoruz. Bu da OK bu arada, fingerprinting bu zaten, ve haklısınız, sarı nokta sayısını arttırırsanız isabetlilik artacaktır.

Fakat "ben nerdeyim" problemine de sadece küçük bir modelleme eklentisi ile geçebiliriz --> en yakın sarı nokta hangisiyse ona "snap" edelim demeyip, sarı noktalara uzaklıklarımızın bir fonksiyonu olarak grid'deki yerimizi bulabiliriz. Bunu fingerprinting ile model-based arasında birşey yapmak gibi düşünebiliriz. O bahsettiğim "fonksiyon" modelimiz oluyor. Mesela (yanlış olduğunu bilsek de) RSS-distance arasında linear bir model varsayarsak, şöyle bir örnek düşünülebilir:

**Evaluate this example numerically:**

1- I'll use a piecewise linear fit (we know it's wrong due to the $1/r^2$ rule, but let's do it anyhow),

I have the following distances for each AP:

AP@(0,0) --> 0.00m --> 1.00 dBm

AP@(0,0) --> 6.32m --> 0.16 dBm

AP@(0,0) --> 7.07m --> 0.14 dBm

... so on and so forth for other APs

2- For AP@(0,0), RSS=0.35 dBm corresponds to 4.89m with the linear fit.

AP@(6,2), RSS=0.31 dBm --> ...

AP@(1,7), RSS=0.20 dBm --> ...

3,4 - ... (see above)

Hesap tam tutmuyordur eminim hata yapmışımdır sayılarda. AP@(0,0) dışındaki AP'lerin yeşil circle'larını rastgele yerlere çizdim, ama fikri anlamanız adına çizimi paylaşıyorum. Buradaki linear modelin yanlış olduğunu neden biliyoruz --> zaten $1/\text{distance}^2$ kuralı var. Fakat o kural bir süre sonra tutmuyor demiştik. Dolayısıyla biz modelin $1/\text{distance}^2$ de olmadığını biliyoruz. Model ne peki? Bu zaten işin zor kısmı. Bu modeli bulmak için de data-driven (ML / AI) teknikleri kullanabiliriz diye düşünüyoruz vs.

Projenin kalanının tam planını çıkaracak noktaya bence geldik. Bu bahsettiğiniz insan yürümesinin hassasiyetini hesaplama vs. gibi şeyleri bi formalize edelim. Bunun için yarın toplansak iyi olur, ama benim 11:00'da malesef müsaitliğim yok yarın. Whatsapp'tan saat belirlemek için ayrıca yazıyorum.

Selamlar,
Burak

From: BURAK SONER <bsoner16@ku.edu.tr>
Sent: Thursday, March 7, 2024 7:57 PM
To: Burak Soner <BSONER@ku.edu.tr>
Subject: Fwd: ELEC491: Fingerprint

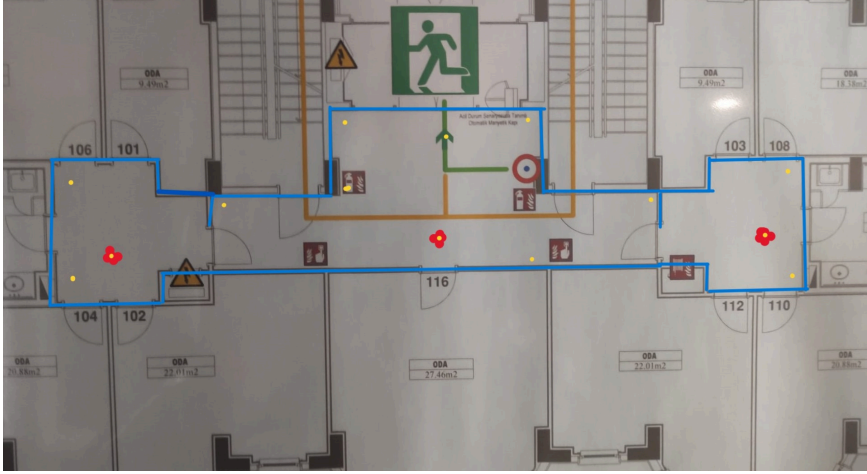
You don't often get email from bsoner16@ku.edu.tr. [Learn why this is important](#)

----- Forwarded message -----

From: **ERDEM ERGEN** <eergen20@ku.edu.tr>
Date: Thu, Mar 7, 2024 at 7:41 PM
Subject: ELEC491: Fingerprint
To: BURAK SONER <bsoner16@ku.edu.tr>, BORA CARPINLIOGLU <bcarpinlioglu21@ku.edu.tr>
Cc: ABDULKADIR BILGE <abilge20@ku.edu.tr>

Hocam Selamlar

Fingerprint metodunu kolaylık sağlaması açısından yurt binasında denemek istedik hem ağa bağlı cihaz sayısı az hemde etrafta gezen kişi sayısı az. Uygulama yaptığımız alanın krokisi :



Resimdeki mavi çizgiler konumlandırma yaptığımız alanın sınırlarını belirtmektedir. Kırmızı noktalar modem olduğu noktalar. Sarı noktalar ise bizim fingerprint için veri aldığımız noktalar. Alanı gridleyip koordinat düzlemine aktarabilmek için yerdeki fayansları kullandık her bir fayans 1 birim olacak şekilde. Farklı noktalarda algoritmayı test ettik. Datapoint az olduğundan çok hassas sonuçlar elde edemedik fakat algoritma yine de beklediğimizden daha iyi sonuçlar verdi.

Datapoint sayısını arttırdığımızda daha hassas sonuçlar vereceğini düşünüyoruz. Data toplarken ve test yaparken alanda kimsenin olmamasına dikkat ettik. İnsanların olduğu durumda test yapmadık yani o durumda da doğru sonuç alıp alamayacağımızı henüz bilmiyoruz. Herhangi bir noktadan alınan RSS değerleri ile her bir noktadaki RSS değerleri arasındaki uzaklığı (euclidean norm) hesaplayıp, en küçük değere sahip olan noktayı anlık konum olarak belirledik. Gözlemlerimizi bir tablo haline getirip ekledim. 6 denemenin 5' inde en yakın noktayı belirledi. Data Pointlerin artırılması durumunda çok daha hassas sonuçlar gözlemleyebiliriz.

Actual Location	Observed Location
-6, 4	6,7
7, -2	14,2
0,7	-6,7
0,0	-6,4
25, 0	28,4
-25, -2	-28,4

Sistemi canlı bir şekilde yapar hale getirmek içinde henüz aklımızda bir metod yok fakat python ile yapılabilen örnekler gördük. Python üzerinde katın bir mapini çizip anlık konumu da mavi bir nokta ile bildirebileceğimiz düşünüyoruz.

Saygılarımla
Erdem Ergen

--
Burak Soner