

Gömülü Sistem Üzerinde Harita Fonksiyonlarının Performans Analizi

G. Çiğdem Çavdaroğlu¹, Erdem Ergen²

¹ Analiz ve Tasarım Lideri, KoçSistem İstanbul Türkiye
cigdem.cavdaroglu@kocsistem.com.tr

² Yazılım Mimarı, KoçSistem İstanbul Türkiye
erdem.ergen@kocsistem.com.tr

Özet. Sunulan çalışmada, Avrupa Birliği Akıllı Trafik Sistemleri aksiyon planı ve Hedef 2023 Ulusal Aksiyon Planını destekleyen Akıllı Trafik Sistemlerine yönelik geleceğin hizmetlerinin yaratılması için ihtiyaç duyulacak uygulama ve donanımların geliştirilmesini adresleyen CoMoSeF projesi kapsamında geliştirilen harita fonksiyonlarının gömülü sistem üzerindeki performans analizleri incelenmektedir. Projede, araçların diğer araçlar ile ve yol kenarı üniteleri ile haberleşmesi sağlanarak elde edilen verilerden sürüş kalitesini ve güvenliğini arttıracak ve trafik yönetimini destekleyecek bilgi sağlanması gerçekleştirilmektedir. Proje kapsamında geliştirilen ve sürücü güvenliği ile rahatlığını artırıcı nitelikteki harita fonksiyonları, Unex firmasına ait Obe-102 isimli donanım üzerinde çalıştırılmaktadır. Bu donanım, 802.11p protokolü üzerinden ETSI standartlarına uygun olarak ITS iletişim fonksiyonlarını sağlamaktadır.

Anahtar Kelimeler. Gömülü sistem, harita fonksiyonları, performans analizi.

1 Giriş

Bu çalışma Eureka [1] kümesi olan Celtic-Plus [2] programı kapsamında, 2012 – 2015 yılları arasında yürütülen “Akıllı Trafik Sistemlerine Yönelik Geleceğin İşbirlikçi Mobil Hizmetleri” (Co-operative Mobility Services of the Future) [3] isimli projede gerçekleştirilen Ar-Ge faaliyetleri kapsamında oluşturulmuştur. Proje çalışmanın kalan kısmında CoMoSeF ismiyle anılacaktır. CoMoSeF projesiyle Avrupa Birliği Akıllı Trafik Sistemleri aksiyon planı [4] ve Hedef 2023 Ulusal Aksiyon Planını [5] destekleyen Akıllı Trafik Sistemlerine yönelik geleceğin hizmetlerinin yaratılması için ihtiyaç duyulacak uygulama ve donanımların geliştirilmesi adreslenmiştir. Bu kapsamda pazar ihtiyaçları çerçevesinde hizmet ve iletişim çözümlerini içeren iş modelleri yaratılmıştır. Projede, araçların diğer araçlar ile ve yol kenarı üniteleri ile haberleşmesi sağlanarak elde edilen verilerden sürüş kalitesini ve güvenliğini arttıracak ve trafik yönetimini destekleyecek bilgi sağlanması gerçekleştirilmiştir. Bu verilerin işlenmesi ile elde edilen bilgiler mobil

uygulamalar ve servisler aracılığı ile son kullanıcı olan sürücüler, yayalar ve trafik otoritelerinin hizmetine sunulmuştur. Projenin Türkiye Konsorsiyu-mu tarafında yürütülen çalışmalarda İstanbul'da ve Sakarya'da bir pilot proje gerçekleştirilmektedir. Projeye 9 ülkeden 23 iş ortağı katılmıştır. Proje kapsamında gerçekleştirilen kullanım senaryolarında; yol kenarı ünitesinden gelen verilerin araç içi donanımda işlenerek uyarı mesajlarının üretilmesi ve üretilen uyarı mesajlarının gösterilmesiyle sürücünün bilgilendirilmesi amaçlanmıştır.

1.1 Proje Kapsamı, Donanım Bileşenleri ve Teknolojiler

KoçSistem, CoMoSeF projesi kapsamında, araç içi cihazlarda bulunacak gömülü yazılım tasarımlarını gerçekleştirmiştir. Cihazların birbirleri ile iletişim kurmasını, konu-munu belirlemesini ve araçtaki sensörlerden bilgi almasını sağlayacak, ağ yığınları, modül ve mikroişlemci / ürün kartı yazılımlarını hazırlamıştır. Akıllı ulaşım sistemleri için oluşturulan 802.11p [6], C2C-CC Network [7], Nmea-183 [8] ve CAN-2.0 [9] yığınları ve kütüphaneleri ile çalışılmış, sağladıkları fonksiyonlar kullanılarak uygulamalar beslenmiştir. Uygulamalar için hayati veri kaynağı olan özelleştirilmiş harita altyapısı mimarisi oluşturulmuştur. Sunulan çalışmada geliştirilen harita fonksiyonlarının gömülü cihaz üzerindeki çalışma performansları incelenmiştir.

Çalışma kapsamında kullanılan gömülü sistem, Unex firmasına ait Obe-102 [10] isimli donanımdır. Obe 102 akıllı ulaşım sistemleri iletişim modülü, diğer endüstri tipi bilgisayarlardan farklı olarak; 802.11p protokolü uyumlu kablosuz veri haberleşmesi yapabilen iletişim donanımı ile bu donanım üzerinde koşan ETSI [11] akıllı ulaşım sistemleri standartlarını destekleyen protokol yığını bulundurmaktadır. Obe 102, 802.11p protokolü üzerinden ETSI standartlarına uygun olarak ITS [12] iletişim fonksiyonlarını sağlamaktadır. Tablo 1'de Obe 102'ye ilişkin genel ve teknik özellikler Şekil 1'de donanım görülmektedir.

Tablo 1. Obe 102 Özellikler

Özellik	Değer
Boyut:	20 x 12 x 3 cm
İşlemci:	Freescale MPC5121e
Anten:	5dBi Omni 5.9GHz DSRC
Sistem Hafızası:	64MB NOR flash, 512MB DDR2 SDRAM
İşletim Sistemi:	Linux Kernel 2.6.32
Standartlar:	IEEE 802.11p, ETSI TS 102 637-2, ETSI TS 102 637-3, ETSI TS 102 636-5-1, ETSI ITS 102 636-6-1, ETSI TS 102 636-4-1, ETSI ES 202-663
SDK:	Toolchain ve API dokümantasyonu
Sistem Servisleri:	SSH, Telnet, HTTP



Şekil. 1. Obe 102 Genel Görünüm

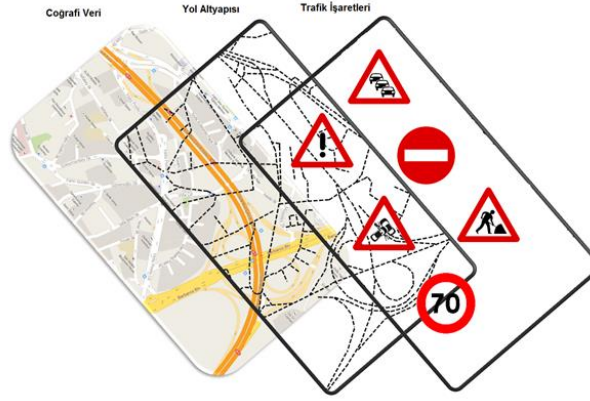
1.2 Literatür

Günümüzde sıklıkla kullanılan birçok uygulama, harita verisi ve harita fonksiyonlarına ihtiyaç duymaktadır. Mekansal veri kullanan bu uygulamalar, masaüstü bilgi-sayarlar, dizüstü bilgisayarlar, sunucular, mobil cihazlar ve ihtiyaca yönelik olarak geliştirilmiş özel donanımlar gibi birçok farklı türde ve kapsamda cihazlarda kullanılmaktadır. Buna bağlı olarak, harita verilerinin sistem performansını en az etkileyecek düzeyde saklanması, sorgulamalarda en hızlı cevabı alabilecek şekilde sunulması birçok farklı araştırmanın konusu olmuştur. Mekansal veri, genel olarak iki farklı yöntemle modellenmektedir. Bu yöntemlerden ilki, “R-tree” gibi mekansal veri yapılarına özel yapıların kullanılmasıdır. Diğer, “B-tree” gibi standart indeksleme tekniklerinin kullanılabilmesi için tek boyutlu uzayda ifade edilen mekansal konum nesnelerinin kullanılmasıdır [18]. Bu yöntemler çerçevesinde geliştirilmiş veri yapılarının her biri, bazı noktalarda yüksek performans gösterirken, bazı noktalarda yetersiz kalmaktadır [19]. Bu nedenle yöntem kullanılabilecek donanım ve problem göz önünde bulundurularak karar verilmesi gerekmektedir. Sunulan çalışmada, mevcut yöntemlere göre modellenen harita verisi, çalışma anında düşük performans göstermiştir. Performansın artırılabilmesi ve daha yüksek doğrulukta çözüm elde edilebilmesi için model üzerinde düzenlemeler yapılarak, probleme yönelik özgün bir çözüm önerisi yapılmıştır.

1.3 Harita Modülü Gereksinimi

Mobil teknolojilerin ve araç teknolojilerinin gelişmesiyle birlikte, harita ve konumlama uygulamaları, anlık kaza bilgileri, trafik sıkışıklıkları, hava durumu, tehlike durumları gibi dinamik parametrelerle araç sürücülerine erken uyarılarda ve bildirimlerde bulunabilmektedirler. Bu uygulamalar, harita statik verilerinin, dinamik katmanlarda yer alan değişken veriler ile birlikte kullanılmasıyla ileri seviyede karar alabilmektedirler. Statik harita verileri, bu senaryoların gerçekleştirilmesi için gerekli

olan dinamik verileri içermemektedir. Bu ek işlevlerin sağlanabilmesi için harita verileri ve konumlama sistemleri ek özelliklere ihtiyaç duymakta, yeni yapılar ile donatılmaları gerekmektedir. Örnek olarak, bir trafik kazasının ya da beklenmedik trafik sıkışıklıklarının sürücüyü bildirilmesi için, harita statik verilerinin yanı sıra harita üzerinde konumlandırılacak dinamik verilere gereksinim duyulur. Bu gereksinimlerin karşılanabilmesi için CoMoSeF projesi kapsamında Yerel Dinamik Harita mimarisi kurulmuştur ve gereksinimleri karşılayacak fonksiyonlar geliştirilmiştir. Yerel Dinamik Harita, objelerin tip, konum ve diğer karakteristik özelliklerinin depolanmasını, güncellenmesini ve ileri aşamalarda işlenmek üzere yeniden bu verilerin sistemden alınmasını, analiz edilmesini, bazı olasılıkların hesaplanmasını, kaza olasılıklarının tahminlenmesini, tehlike yaratan objelerin tespit edilmesini sağlar [13]. Otomatik sürüş sistemleri ve erken uyarı sistemleri için gerekli olan veriler, araç sensörleri aracılığıyla alınır; tüm trafik ağıyla paylaşılır; buna ek olarak statik sayısal harita verilerine dinamik bir katman olarak da eklenebilir. Şekil 2’de örnek bir Yerel Dinamik Harita veri yapısı görülmektedir.



Şekil. 2. Yerel Dinamik Harita

1.4 Harita Fonksiyonları

Proje kapsamında gerçekleştirilen senaryolarda harita modülü, araca ait konum bilgisi ile tetiklenir. Diğer fonksiyonlar, aracın harita üzerinde konumlandırılmasından sonra çalışmaktadır. Proje kapsamında gereksinim duyulan harita fonksiyonları, aracın konum bilgisi baz alınarak harita verisi üzerinde konumlandırılması, iki konum bilgisi arasındaki yol verisi üzerinden mesafe değeri, anlık olay bilgilerinin dinamik katmanlara eklenmesi ve aracın konum bilgisi baz alınarak seyir yönüne göre yolu üzerinde yer alan olay bilgilerinin dinamik katmanlardan sorgulanmasıdır. Tüm harita fonksiyonları, öncelikle aracın harita üzerinde konumlandırılmasını gerektirmektedir.

1.5 En Yakın Yol Bulma ve En Kısa Mesafe Hesaplama Problemleri

Konum bilgisi baz alınarak, bir yol veri yığını içerisinde en yakın yol bilgisinin bulunması ve iki konum arasındaki en kısa mesafenin hesaplanması literatürde sıklıkla karşılaşılan ve üzerinde çalışılan problemlerden birisidir. Bu problemler için kullanılan yöntemler, yol veri yığını içerisinde en kısa mesafe hesabı, en yakın komşuluk (NN), Voronoi ağında en yakın komşuluk (VN3), A* algoritması, Dijkstra algoritması, D* algoritması olarak sıralanabilir [14, 15]. Oyun algoritmalarında en yakın noktanın hesaplanması konusunda yapılmış bir çalışma sonucunda elde edilmiş en kısa mesafe hesaplama algoritmalarının çalışma performanslarının karşılaştırması Tablo 2’de görülmektedir [15].

Tablo 2. En Kısa Yol Algoritmalarının Karşılaştırılması

Algoritma	Ortalama Düğüm Ziyareti	Bir Patika Üzerinde Ortalama Düğüm Ziyareti	Hesaplama Süresi (Saniye)
A*	23.915	2.900	3.564
Dijkstra	145.665	2.855	3.507
D*	3.640	2.770	3.472

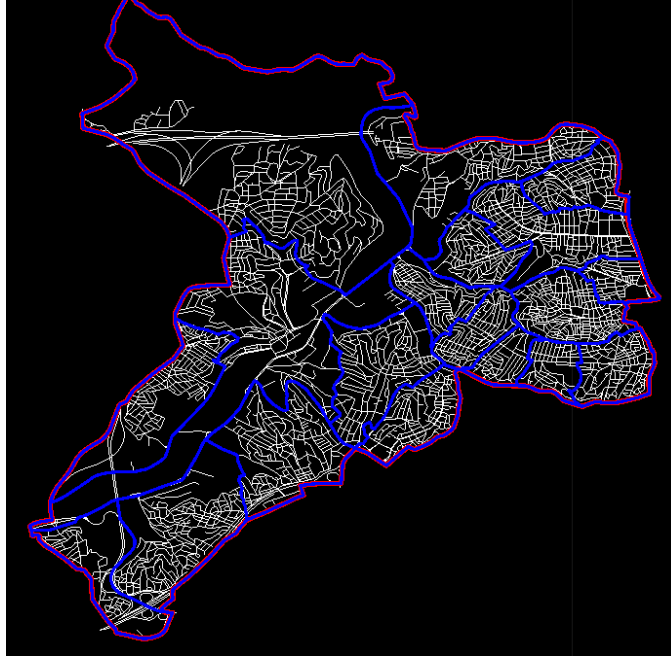
2 Yöntem

Harita verileri birisi XML birisi yazı formatı olmak üzere iki farklı dosya yapısında saklanmıştır. Bilgisayar ortamında yapılan testlerde XML dosyaları, Obe – 102 donanımı üzerinde yapılan testlerde yazı dosyası kullanılmıştır. Gereksinim duyulan harita verisi sadece yol verileri, mahalle ve ilçe sınırlarıdır. Bu nedenle oluşturulan dosya yapılarında diğer coğrafi bileşenler saklanmamıştır. Yol verileri ID değerlerine göre sıralı olarak saklanmıştır. Şekil 3’te dosya yapıları görülmektedir.

XML Dosya Yapısı	Yazı Formatlı Dosya Yapısı
<pre><polyline> <GisType>Road</GisType> <id>4</id> <layerid>0</layerid> <points>X=28.981053, Y=41.100201, Z=0;X=28.9 <GisId /> <GisName /> <GisAddress /> </polyline> <polyline> <GisType>Road</GisType> <id>5</id> <layerid>0</layerid> <points>X=28.981498, Y=41.100423, Z=0;X=28.9 <GisId /> <GisName /> <GisAddress /> </polyline></pre>	<pre>41.089528 28.984681 41.089718 28.984983 27 19 16 41.102614 28.983290 41.102642 28.982354 28 30 31 41.102642 28.982354 41.102427 28.981529 29 31 32 41.102427 28.981529 41.102130 28.980977 30 32 33 41.102130 28.980977 41.101780 28.980670 31 33 34 41.101780 28.980670 41.101378 28.980453 32 34 35 41.101378 28.980453 41.100663 28.980325 33 35 36 41.100663 28.980325 41.100332 28.980289 34 36 8 41.100332 28.980289 41.100148 28.980276 35 8 3 41.100148 28.980276 41.099903 28.980308 36 3 37 41.099903 28.980308 41.099782 28.980351 37 37 38 41.099782 28.980351 41.098538 28.981109 38 38 39 41.098538 28.981109 41.096433 28.981811 39 39 40 41.096433 28.981811 41.095647 28.982518 40 40 15 41.095647 28.981811 41.091519 28.985149 41 13 41 41.091519 28.985149 41.092735 28.985135 42 41 42</pre>

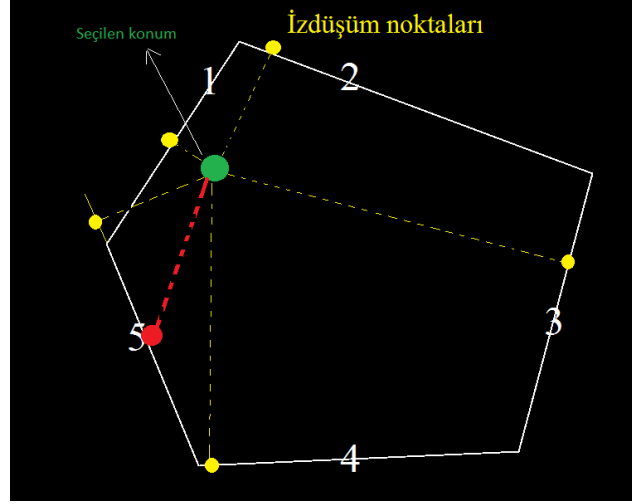
Şekil 3. Dosya Yapıları

Proje pilot çalışmalarında İstanbul - Kağıthane harita verileri kullanılmıştır. Kağıthane bölgesi haritasının genel görünümü Şekil 4’te görülmektedir.



Şekil. 4. Test Verisi, İstanbul Kağıthane

Yol verileri uygulama çalışma anında dosyadan okunarak belleğe yüklenmektedir. Aracın konum bilgisine göre harita üzerinde konumlandırılması için, konuma en yakın uzaklıkta bulunan yolun bulunması gerekli olmuştur. Araç konumu, harita verisi içerisinde bir nokta olarak, yol verisi de bir çizgi olarak ele alındığında, problem bir noktanın en yakın olduğu çizginin bulunmasına dönüşmektedir. Bir nokta-nın bir çizgiye olan en kısa uzaklığı, noktadan çizgiye dik olarak düşülen çizginin uzunluğu kadardır. Bu nedenle araç konumuna ilişkin noktadan harita verisi içerisinde yer alan tüm yol çizgilerine dik çizgiler düşülerek nokta iz düşürülmüş ve elde edilen çizginin uzunluk değerleri hesaplanmıştır. Enlem – boylam değeri olarak tutulan konum verileri arasındaki uzunluklar hesaplanırken Haversine algoritması kullanılmıştır [16]. Belirli bir konum verisine göre elde edilen iz düşüm noktaları Şekil 5’te görülmektedir.



Şekil 5. En Yakın Yol Hesabı

Nokta çizgiye iz düşürüldüğünde elde edilen iz düşüm noktası çizgi üzerinde yer alı-yorsa doğrudan iz düşüm çizgisinin uzunluk değeri hesaplanmıştır. Ancak iz düşüm noktası yol çizgisinin dışında yer alıyorsa bu durumda yol çizgisinin orta noktası hesaplanmış ve orta nokta ile konum noktası arasındaki mesafe gerçek değer olarak kabul edilmiştir. İşlem sonunda elde edilen uzunluk değerlerinden en küçük olan değeri sağlayan yol çizgi verisi aracın üzerinde seyrettiği yol çizgisi olarak kabul edilmiştir. Bu yöntemin uygulanması sırasında veri setinde kaç tane yol çizgisi verisi var ise o değere kadar dönecek bir döngü kurulur ve bahsedilen hesaplamalar gerçekleştirilir. Buna bağlı olarak da yol veri miktarı yöntemin çalışma performansını doğrudan etkilemektedir.

Bu sonuca bağlı olarak harita verilerinin dinamik olarak küçültülmesine ilişkin bir yöntem geliştirilmesine karar verilmiştir. En yakın nokta arama problemlerinde sıklıkla kullanılan dörtlü ağaç yapısı incelenerek bu veri yapısının geliştirilmesine karar verilmiştir. Dörtlü ağaç veri yapısında veri yığını aşamalı olarak dörtlü karelere ayrılır. Adım adım dörtlü kareye ayırma işlemi, bölünecek bir bütün alanın tamamen dolu veya tamamen boş kalmasına kadar sürdürülür. Sonuçta elde edilen tüm kare birimler içerisinde ya sadece bir tane nokta bulunur ya da kare birim tamamen boştur [17]. En kısa yolun hesaplanması sırasında ise sadece en kısa mesafede bulunan noktanın bulunması yeterli olmaz, bunun yerine en yakın belirli bir mesafede bulunan tüm yol çizgilerinin ele alınmasıyla noktanın yol çizgilerine olan dik uzaklıkları hesaplanmalı ve en kısa mesafeyi üreten yol çizgisi seçilmelidir. Bu nedenle dörtlü ağaç yapısı değiştirilerek proje kapsamında kullanılan harita veri yapısı ve fonksiyonlarının gereksinimlerini karşılayacak hale getirilmiştir.

Veri yapısının belleğe yüklenmesi sırasında tüm veriyi kapsayan bir alan oluşturmak üzere en küçük ve en büyük enlem ve boylam değerleri hesaplanmıştır. Ağaç

yapısının kaçı oluşturulacağı bilgisi dinamik olarak verilmektedir. Test çalışmaları sırasında kullanılan test verisi için bu değer sabitlenmiş ve 5 olarak verilmiştir. Veri bölgesi bu değere göre karelere ayrılarak her bir yol noktasının hangi bölge içerisinde kaldığı bilgisi bulunmuş ve saklanmıştır. Bir konum noktası için sorgu yapılması gerektiğinde tüm yol verilerine olan uzaklık değerlerinin hesaplanması yerine öncelikle konumun hangi bölge içerisinde kaldığı belirlenmiş ve sadece bu bölge içerisinde kalan yol verilerine olan uzaklıklar hesaplanmıştır. Yol verilerinin alt bölgelere ayrılmış hali Şekil 6'da görülmektedir.



Şekil. 6. Harita Alt Alanlı Ağaç Yapısı

3 Sonuçlar

Bilgisayar ortamında yapılan testlerde 10.000 adet yol verisi işlenebilmiş ve fonksiyonlar uygulanabilmiştir. Ancak Obe 102 üzerinde gerçekleştirilen testlerde yöntem başarısız olmuştur. Bunun üzerine geliştirilen yeni harita veri yapısı (harita alt alanlı ağaç yapısı) yöntemi ile yol veri miktarı küçültülmüş ve mahalle bazında çalışma-ya devam edilmiştir. Tablo 3'te farklı yol veri miktarlarına göre ortalama CPU değerleri gösterilmiştir.

Tablo 3. Ortalama CPU Değerleri

Yol Veri Sayısı (Adet)	Ortalama CPU
1902	40.5
409	29.65
73	8.95

Tablodan görüleceği üzere, veri miktarının azalmasıyla birlikte Obe 102 üzerinde elde edilen performans değerleri oldukça büyük oranda artış göstermiştir. Performanstaki artış büyük oranda veri miktarının azaltılmasına dayanmaktadır. Yöntemin uygulanmasıyla veri miktarı $1/(\text{bölge sayısı})^2$ oranında azaltılmıştır.

Akıllı Ulaşım Sistemleri konusu, ülkemizde ve Avrupa’da son yıllarda çok fazla çalışma yapılan ve yeni gelişmekte olan konulardan birisidir. Bu alanda henüz ürün niteliğine gelmiş bir çalışma bulunmamaktadır. Standartların da henüz net olarak oluşmamış olması, standartlaşma sürecinde, mekansal veri modellemesi konusunda katkıda bulunabilmeyi mümkün kılmaktadır. Bu süreçte yapılacak olan araştırma çalışmalarında, önerilen veri modeli kullanılabilir. Ancak ürünleşme sürecinde, uygulama birçok yeni fonksiyona ve daha yüksek çalışma performansına gereksinim duyacaktır. Bu geliştirmelerle, önerilen veri modelinin iyileştirilmesi ya da geliştirilmesi öngörülmektedir.

4 Tartışma

Yöntemin testleri sırasında sadece belirli bir bölgeye ilişkin test verileri kullanılmıştır. Yöntemi genelleştirmek ve bütün bir harita sisteminin içerisine entegre etmek gerekli olduğunda, tüm harita verilerini dinamik olarak bölgelere ayıracak ve kaç alt alan oluşturulması gerekli olduğunu hesaplayacak ara modüller geliştirilmesine gerek bulunmaktadır. Test çalışmaları kapsamında alt alan sayıları sabit olarak verilmiştir, ancak bu değerlerin başka bir bölgeye ilişkin harita veri yapısında doğrudan kullanımı uygun olmayacaktır. Harita verisi içerisinde yer alan yol verilerinin alt alanlara uygun bir şekilde dağılımını sağlayacak yardımcı yöntemler geliştirilmesi gereklidir. Projenin devamında bu yöntemlerin araştırılmasına devam edilecektir.

5 Teşekkür

Çalışmanın test süreci, İSBAK ve Otokar firmaları ile birlikte gerçekleştirilmiştir. Test verileri İSBAK firması tarafından sağlanmıştır. İSBAK [] ve Otokar [] firmalarına işbirlikleri için teşekkürlerimizi sunarız. Geliştirme sürecindeki işbirlikleri ve çalışmalarından ötürü tüm CoMoSeF proje ortaklarına da teşekkürlerimizi sunarız.

Kaynaklar

1. <http://www.eurekanetwork.org/projects>
2. <https://www.celticplus.eu/>
3. <http://www.comosef.eu/>
4. http://ec.europa.eu/transport/themes/its/road/action_plan/
5. http://www.ubak.gov.tr/BLSM_WIYS/UBAK/tr/AUF/AUS_Strateji_Belgesi_Eki_Eylem_Planı.pdf
6. http://en.wikipedia.org/wiki/IEEE_802.11p
7. https://portal.etsi.org/CTI/Downloads/TestSpecifications/tr_102698v010101p.pdf
8. http://en.wikipedia.org/wiki/NMEA_0183
9. http://en.wikipedia.org/wiki/CAN_bus
10. www.unex.com.tw/m/download/file/fid/525
11. <http://www.etsi.org/>
12. <http://www.etsi.org/technologies-clusters/technologies/intelligent-transport>
13. Local Dynamic Maps for Cooperative Systems, SAFESPOT Innovative technologies – SINTECH SubProject, C., Bartels, Strategic Research Project Manager, ITS World Congress, London, 2006.
14. Fast Nearest Neighbor Search on Road Networks, Y. Ioannidis vd, Springer-Verlag Berlin Heidelberg 2006.
15. Comparison of Efficiency in Pathfinding Algorithms in Game Development, N. Krishnaswamy, De-Paul University, Technical Reports College of Computing and Digital Media, 2009.
16. http://en.wikipedia.org/wiki/Haversine_formula
17. <http://en.wikipedia.org/wiki/Quadtree>
18. An Introduction to Spatial Databases, R. H. Gutting, VLDB Journal v3, n4, 1994.
19. Spatial Data Structures, H. Samet, Computer Science Department and Institute of Advanced Computer Studies and Center for Automation Research University of Maryland, Appears in Modern Database Systems: The Object Model, Interoperability, and Beyond, W. Kim, ed., Addison Wesley/ACM Press, Reading, MA, 1995.