

Üç Boyutlu Konumsal Verinin Web Tabanlı Yönetiminde, OGC Standardı 3D Tiles'in Hiyerarşik Veri Yapılarıyla Gerçekleştirimi Implementation of the OGC Standard 3D Tiles with Hierarchical Data Structures in Web Based Management of 3D Spatial Data

Ziya Usta^{1*}, Çetin Cömert¹

¹Artvin Çoruh Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, 08100, Artvin/Türkiye.

²Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, 61080, Trabzon/Türkiye.

ARAŞTIRMA MAKALESİ

*Sorumlu yazar:

Ziya Usta
ziyausta@artvin.edu.tr

doi: 10.48123/rsgis.1059733

Yayın süreci

Geliş tarihi: 18.02.2022
Kabul tarihi: 02.03.2022
Basım tarihi: 14.02.2022

Özet

3B Kent Modelleri (3BKM) pek çok kent objesinden oluşan yüksek veri hacmine sahip verilerdir. Bu durum 3BKM'lerinin web tabanlı olarak yönetilmesi ve görüntülenmesinde performans sorunları yaratmaktadır. Bu nedenle 3BKM'lerinin bölümlenerek (tiling) daha küçük bölümlere ayrılması gerekmektedir. 3D Tiles büyük boyutlardaki 3B veri kümelerinin web üzerinden transferi için tasarlanmış bir spesifikasyon ve OGC standardıdır. 3D Tiles'in temeli, yalnızca görüntüde görünecek olan bölümlerin web üzerinden transfer edilmesini sağlayan hiyerarşik detay seviyelerine dayanmaktadır. 3BKM'lerinin 3D Tiles spesifikasyonuna göre bölümlenmesini sağlayan herhangi bir açık kaynak kodlu (AKK) web tabanlı yazılım bileşeni bulunmamaktadır. Bu çalışmada 3D Tiles spesifikasyonuna uygun olarak 3B bölümlenme yapan web tabanlı açık kaynak kodlu (AKK) yazılım bileşeni geliştirilmiştir. Bölümlenme için R-Tree ve Adaptive QuadTree hiyerarşik veri yapıları kullanılmıştır. Üretilen 3B bölümler Cesium.js AKK JavaScript kütüphanesi kullanılarak görselleştirilmiştir. Adaptive Quadtree ve R-Tree veri yapıları 3B bölümlenme açısından irdelenmiş, oluşturulma süreleri, veri güncelleme süreleri ve konumsal sorgu performansları karşılaştırılmıştır. Böylelikle geliştirilecek uygulamanın genel özelliklerine göre hangi veri yapısının seçilmesi gerektiği belirlenmiştir.

Anahtar kelimeler: 3B CBS, 3B WebCBS, 3D Tiles, Hiyerarşik veri yapıları

Abstract

3D City Models (3DCMs) consisting of many city objects are huge in size. This situation causes performance problems in web-based management and displaying of 3DCMs. 3DCMs must be decomposed to smaller data chunks called "tiles". 3D Tiles is a specification and OGC standard designed for transferring large 3D datasets over the web. 3D Tiles is based on hierarchical detail levels that allow only the parts that will appear in the image to be transferred over the web. There is no open-source web-based component for tiling 3DCMs according to 3D Tiles. In this study, a web-based 3D Tiling component that tiles 3DCMs according to 3D Tiles has been developed. R-Tree and Adaptive QuadTree hierarchical data structures have been used to decompose 3DCM. Generated 3D tiles have been rendered using open source Cesium.js JavaScript library. Adaptive QuadTree and R-Tree data structures have been investigated in the context of tiling and their performance have been compared in terms of construction times, data update times and spatial query performances. Thus, which data structure should be selected has been determined according to the general characteristics of the application to be developed.

Keywords: 3D GIS, 3D WebGIS, 3D Tiles, Hierarchical data structures

1. Giriş

3B Kent Modelleri (3BKM) artan bir ilgi görmektedir ve web tabanlı yönetimler oldukça faydalıdır. HTML5 ve WebGL gibi teknolojilerin de gelişmesiyle yeni bir araştırma alanı ortaya çıkmıştır “3B WebCBS”. Genel olarak WebCBS uygulamaları geliştirmenin önündeki en büyük zorluk yüksek boyutlu verilerden kaynaklı performans sorunlarının üstesinden gelinmesidir. Bu durum daha fazla performans gerektirdiğinden 3B WebCBS uygulamaları için daha da önemlidir. Bölümleme (Tiling) yaklaşımı veri setini daha küçük parçalara bölerek performansı arttırmak için kullanılan etkin bir yöntemdir. Böylece sadece ihtiyaç duyulan bölümler sunucudan istenerek, transfer edilen veri hacmi azaltılır ve performans arttırılır.

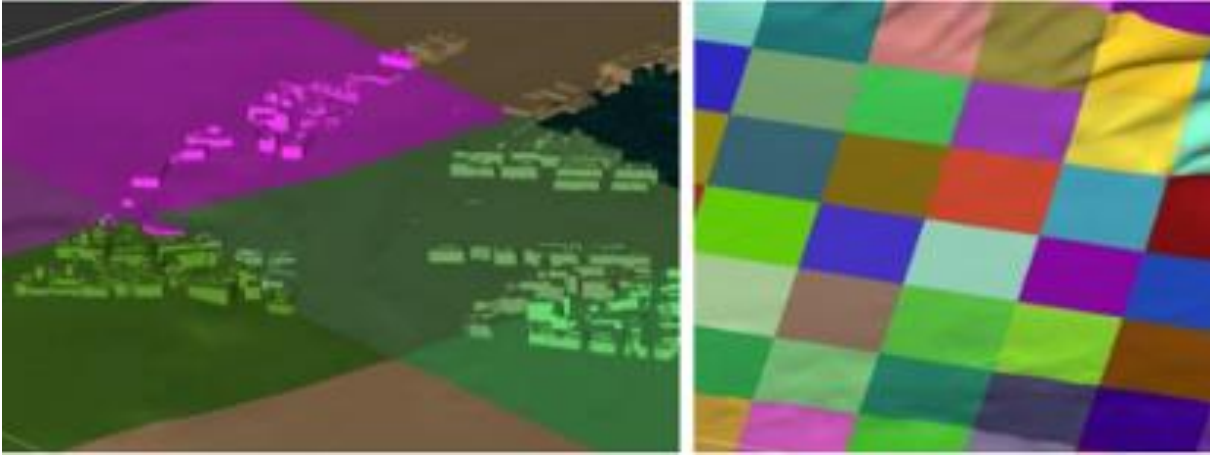
2B vektör ve raster verinin bölünmesi için pek çok çalışma yapılmıştır fakat 3B konumsal verinin bölünmesi nispeten yenidir ve bu konuda yapılan çalışmaların sayısı azdır. 3B verinin bölünmesi 2B verinin bölünmesinde çok daha karmaşıktır.

OGC 3D Tiles, 3B konumsal verinin bölünmesi ve bölünmüş verinin web üzerinden gönderimi için geliştirilmiş bir standarttır. Bu standart 2019 yılı itibarıyla OGC standardı olarak kabul edilmiştir. 3D Tiles, hiyerarşik detay seviyelerine imkan veren konumsal veri yapılarına dayanmaktadır (GitHub, 2021) böylece yalnızca ihtiyaç duyulan bölümler sunucudan istenir. 3D Tiles, R-Tree, QuadTree, Octree, Kd-Tree gibi pek çok veri yapısını desteklemektedir. Bölümlemenin hangi veri yapısına göre yapılacağı tamamen geliştiricilere bırakılmıştır.

OGC 3D Tiles önemli bir standart olmasına karşın, bu standarda göre bölümleme yapan yazılım bileşeni sayısı oldukça azdır. Web tabanlı açık kaynak kodlu bir yazılım bileşeni ise henüz yoktur. Bu çalışmada, 3D Tiles standardına göre bölümleme yapan bir yazılım bileşeni geliştirilerek test edilmiştir.

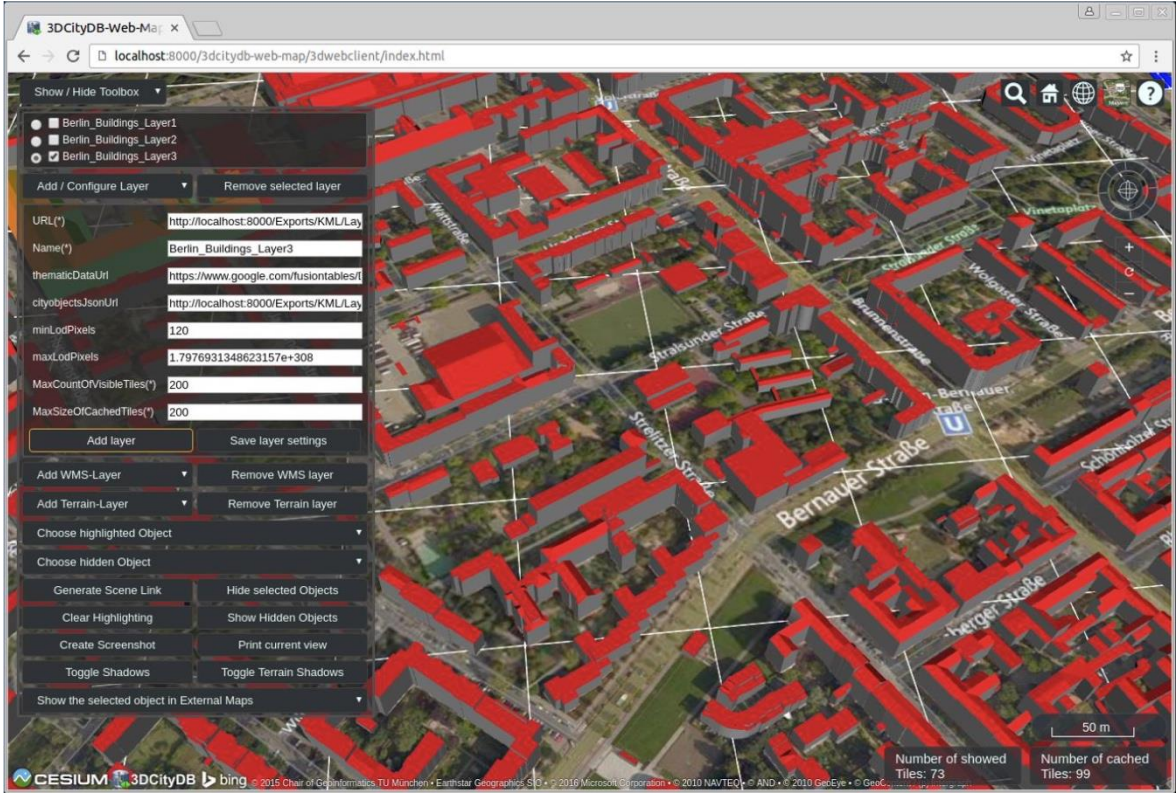
2. Benzer Çalışmalar

Gesquiere ve Manin (2012) bir istemci-sunucu mimarisi geliştirerek CityGML verisinin web tabanlı olarak görselleştirilmesini sağlamıştır. Bu amaçla, CityGML verisini düzenli dikdörtgen bölümlere ayırmışlardır (Şekil 1).



Şekil 1. Kent modelinin düzenli bölümlere ayrılması (Gesquiere ve Manin, 2012).

Prandi vd. (2013) bir akıllı kent platformu geliştirmişlerdir ve bu platform için pek çok web servisi geliştirmişlerdir. Bu amaçla CityGML verisini düzenli dikdörtgen bölümlere ayırmışlardır. Chaturvedi vd. (2015) 3DCityDB yazılımı için bir web istemcisi geliştirmiştir. Bu çalışmada da 3BKM düzenli dikdörtgen bölümlere ayrılmıştır (Şekil 2).



Şekil 2. 3DCityDB yazılımının web istemcisi ve düzenli bölüm yapısı (Chaturvedi vd. 2015)

Gaillard vd. (2015) 3BKM'lerinin web tabanlı görüntülenmesi için bir framework geliştirmişlerdir. Yapılan çalışmada CityGML dosyası JSON'a çevrilmiş ve 3BKM'leri düzenli yapıdaki dikdörtgen bölümlere ayrılmıştır. Bölümlerin tarayıcıda görüntülenmesi için de Three.js JavaScript kütüphanesi kullanılmıştır. Kramer ve Gutbell (2015) çalışmalarında 3B uygulamalar geliştirmiş ve WebGL kütüphanelerini test etmişlerdir. Yaptıkları çalışmalar kapsamında da CityGML verisini düzenli dikdörtgen bölümlere ayırmışlardır.

Buraya kadar bahsi geçen çalışmalar, düzenli bir bölümlendirme işlemi gerçekleştirmiştir. Bölümlendirme hiyerarşik değildir. Böyle bir bölümlendirmede, veri yoğunluğu dikkate alınmamakta, kent modeli eşit uzunluklu dikdörtgen alanlara bölünmektedir. Bu durum heterojen veri boyutlarına sahip bölümlerin oluşmasına yol açmaktadır.

Koukofikis vd., (2018) hiyerarşik veri yapıları kullanarak bölümlendirme gerçekleştirmiştir. 3D Tiles ve 3D Portrayal Service standartları kullanılarak 3B görselleştirme gerçekleştirilmiştir. Bu çalışmada bölümlendirme Cesiumion, FME gibi ticari yazılımlar kullanılarak gerçekleştirilmiştir. Böyle bir iş akışında, birbirleriyle entegre çalışmayan bu yazılımlar pek çok kullanıcı müdahalesini gerektirmektedir.

Gaillard vd. (2020) 3D Tiles standardını kullanarak 3BKM'nin görselleştirilmesi için çok ölçekli çözünürlüğü destekleyen bir metod geliştirmişlerdir. Bu çalışmada bölümlendirme işlemi yol ağı verisi kullanılarak hiyerarşik olarak yapılmış, ana yollara komşu binalar üst bölümlerde, yan yollara komşu binalar alt bölümlerde yer almıştır. Böyle bir bölümlendirmede bölümlendirme veri yoğunluğuna göre yapılmadığından heterojen veri boyutlarına sahip bölümler oluşmaktadır. Ayrıca bölümlerin maksimum veri yoğunluğu kontrol edilebilir değildir.

Lu vd. (2020) yaptıkları çalışmada büyük ölçekli meteorolojik verinin gerçek zamanlı olarak web tabanlı görselleştirmesini gerçekleştirmişlerdir. Bunun için nokta bulutu verisi Octree veri yapısı kullanılarak 3D Tiles standardına göre bölümlenmiştir. Bu çalışma sadece nokta bulutunun bölümlenmesini içermektedir 3BKM'ni kapsamamaktadır.

Xu vd. (2020) yaptıkları çalışmada BIM modellerinin 3D Tiles standardı kullanılarak bölümlenmesini sağlayan bir çalışma gerçekleştirmişlerdir. Bu çalışmada BIM modelleri bölümlenmiştir ve 3BKM'lerini kapsamamaktadır. Ayrıca çalışmada otomatik bir işlem akışı sağlanamamakta, farklı yazılım bileşenleriyle, örneğin IFCShell gibi, farklı işlem adımları gerçekleştirilmiştir ve gerçekleştirilen çözüm web tabanlı değildir.

Akademik çalışmalar dışında yazılım bileşenlerine de bakacak olursak "obj23dtiles" açık kaynak kodlu bir Node.js modülüdür. En büyük eksikliği herhangi bir bölümlendirme gerçekleştirmemesi sadece obj formatındaki tüm kent modelini 3D Tiles'in veri formatı olan tek bir b3dm dosyasına çevirmesidir. Başka bir açık kaynak kodlu Node.js modülü de "citygml-to-3dtiles"tir. Bu modül de herhangi bir bölümlendirme yapmadan CityGML formatındaki veriyi tek bir b3dm dosyasına çevirmektedir.

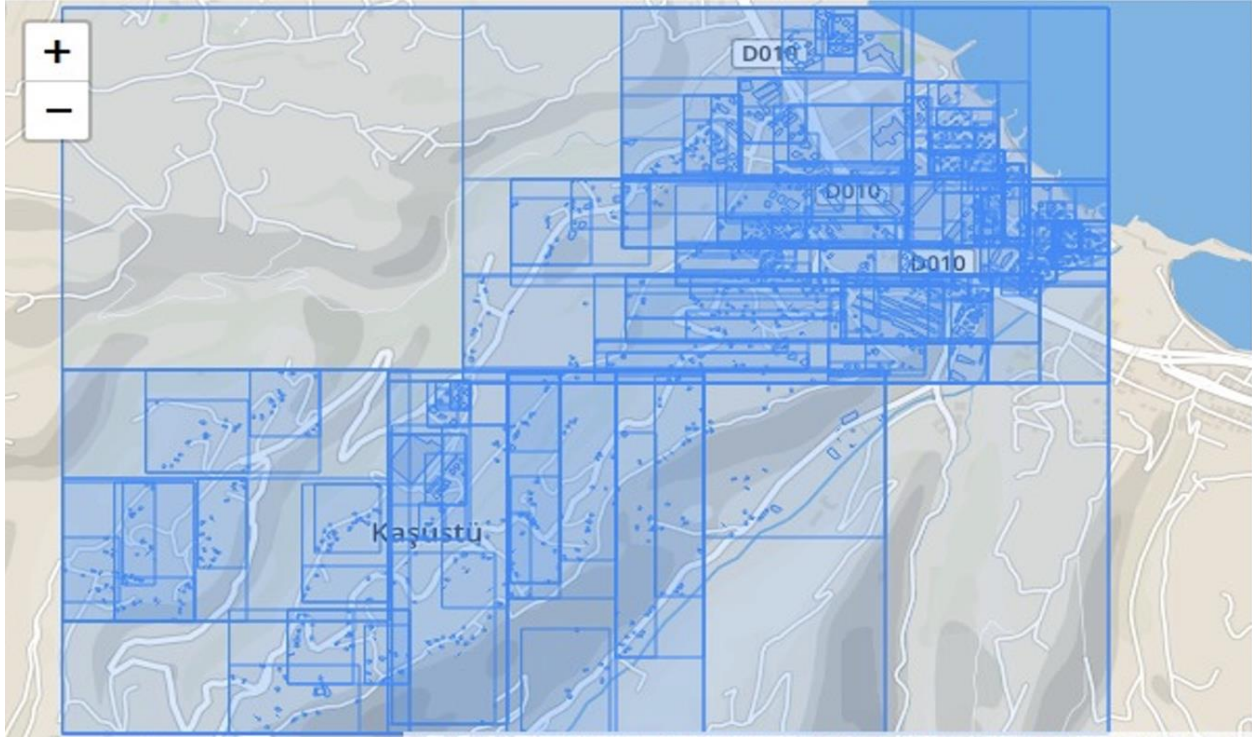
Bu anlamda en gelişmiş açık kaynak kodlu bileşen "py3dtiles"tır. Py3dtiles, CityGML datasını 3D Tiles standardına göre hiyerarşik olarak bölümlenmektedir. En büyük eksikliği ise web tabanlı bir bileşen olmamasıdır. Yapılan bu çalışmayla bu eksiklikler giderilmeye çalışılmıştır.

Literatürdeki çalışmalar ve mevcut yazılım bileşenleri incelendiğinde bazı eksiklikler görülmüştür. İlk olarak, pek çok çalışmada düzenli bir bölümlenme uygulanmış, veri yoğunluğu gözetilerek hiyerarşik bir bölümlenme yapılmamıştır. Geliştirilen pek çok metodoloji pek çok kez kullanıcı müdahalesini gerektirmekte, otomatik olarak çalışmamaktadır. Ayrıca pek çoğu web tabanlı olarak çalışmamaktadır. Üretilen bölümler web tabanlı olarak gösterilmekte, bölümlenmenin kendisi web tabanlı yapılmamaktadır.

3. Yöntem

3.1. R-Tree Veri Yapısı Kullanılarak Bölümlenme

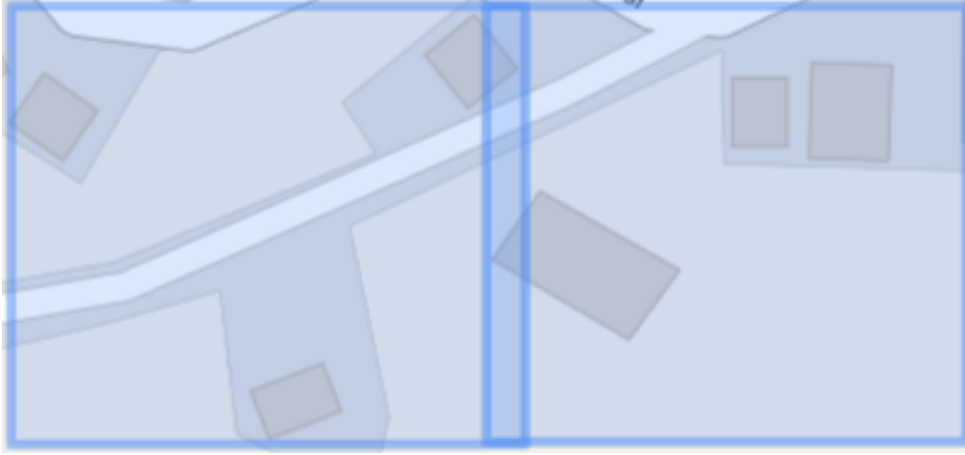
R-Tree, verinin yinelenmeli olarak her bir bölümde istenen veri boyutundan daha fazla veri içermeyene kadar bölümlenmesi ile oluşur. R-Tree aşağıdan yukarı ve yukarıdan aşağı olmak üzere 2 yöntemle oluşturulabilir. Yukarıdan aşağıya metodunda, R-Tree oluşturma işlemi çatı bölümün (root node) 2 alt bölüme bölümlenmesi ile başlar. Bu bölümler birbirine en uzak 2 objeyi içerirler. Daha sonra kalan objeler bu iki bölümden birine dağıtılır. Kalan objelerin hangi bölüme ekleneceği ise minimum hacim genişlemesi ile bulunur. İlgili obje, her iki bölümün sınırlayıcı kutusu ile kıyaslanarak, bu kutulardan hangisinin, objeyi içine alması için daha az genişletilmesi gerekiyorsa, o bölüme eklenir. Bu uygulamada da yukarıdan aşağıya metodu kullanılarak R-Tree oluşturulmuştur (Şekil 3).



Şekil 3. 3B Konumsal verinin bölümlenmesi ile oluşan r-tree yapısı

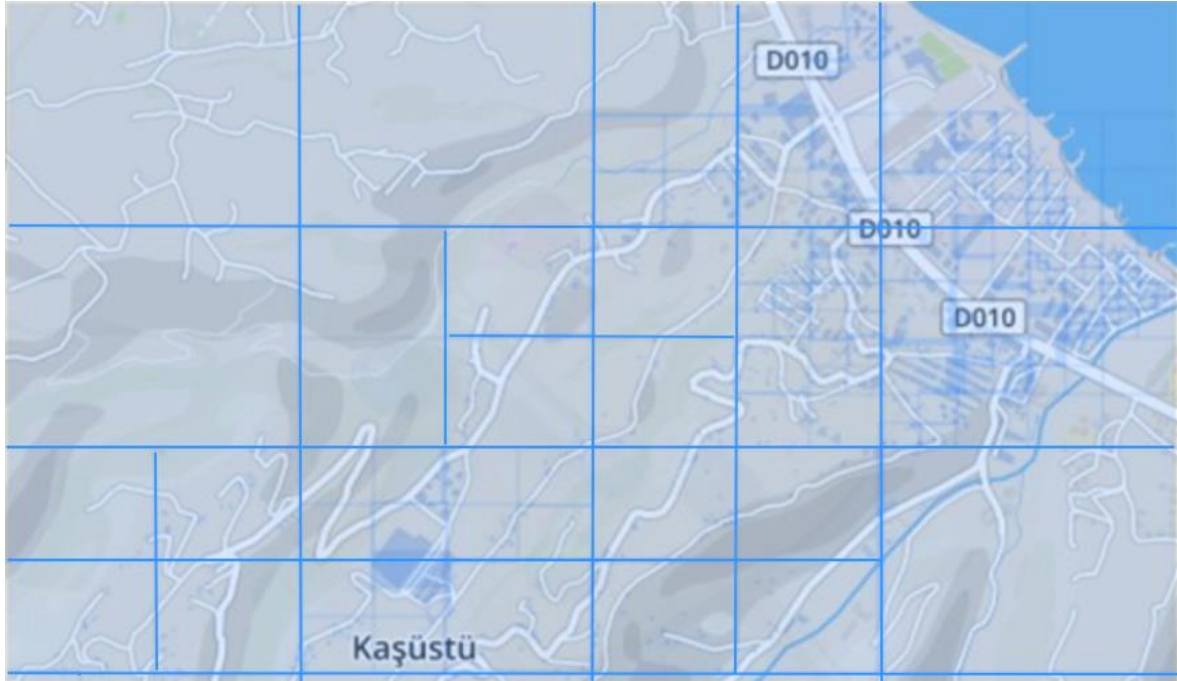
3.2. Adaptive QuadTree Yapısı Kullanarak Bölümlenme

QuadTree, verinin bir veri boyutu sınırlamasına göre 4 eşit alt bölüme yinelenmeli olarak bölümlenmesi ile oluşur. QuadTree, çatı bölümün 4 alt bölüme bölümlenmesi ile başlar. Objeler bu dört bölümden hangisinin içinde yer alıyorsa o bölüme eklenirler. Sonra alt bölümler veri boyutunu aşıyorsa tekrar 4 alt bölüme bölünürler. Bu işlem tüm bölümler veri sınırını aşmayınca kadar devam eder. QuadTree'de bölüm sınırları önceden hesaplandığı için bazı objeler birden fazla bölümün sınırı ile kesişebilir. Bu durumda obje en çok hangi bölümle kesişiyorsa, o bölümün sınırları objeyi içine alacak şekilde genişletilir (Şekil 4).



Şekil 4. 2 Farklı bölümlle kesişen obje için bölüm sınırının genişletilmesi

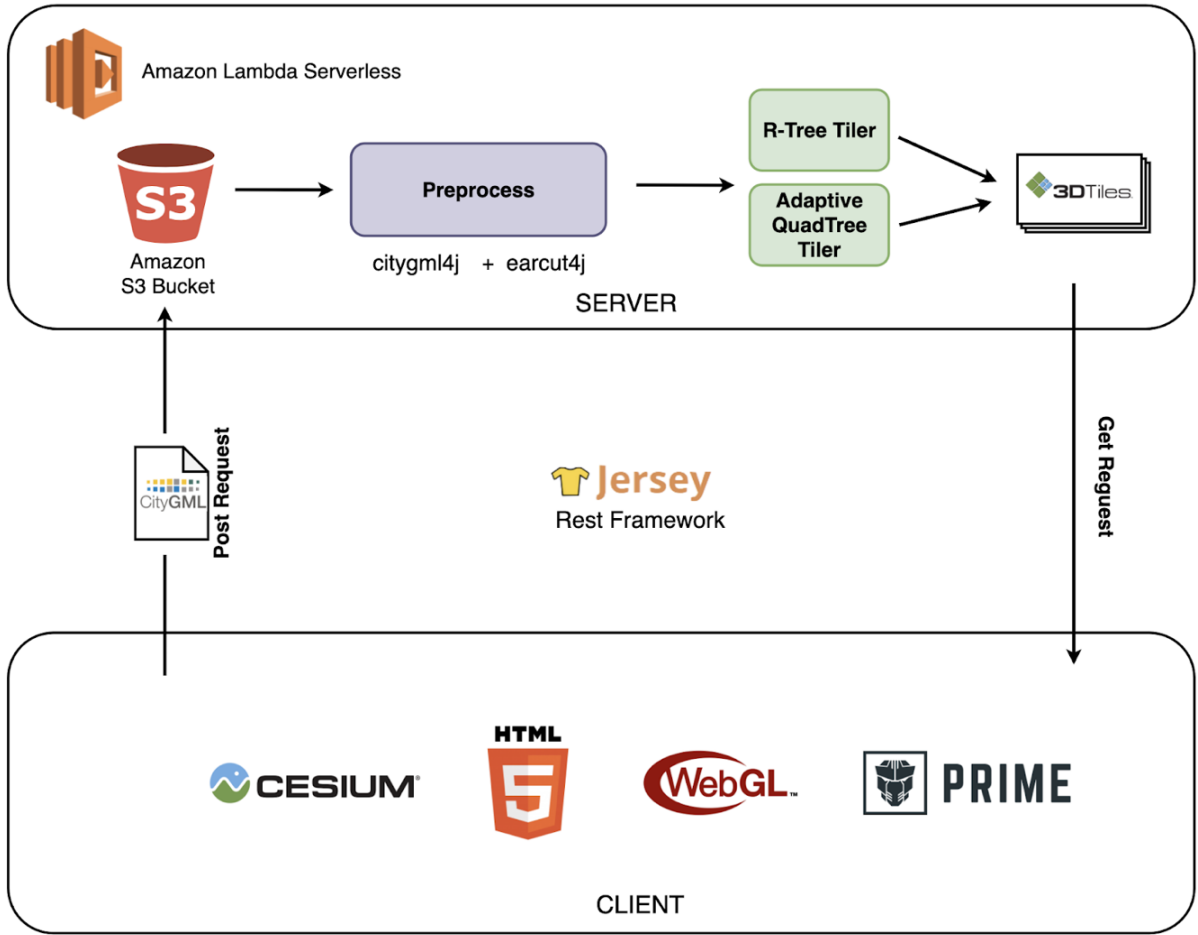
3BKM'nin Adaptive QuadTree veri yapısı ile bölümlenmesi sonucu oluşan yapı Şekil 5'te görülmektedir.



Şekil 5. Adaptive QuadTree

3.3. 3D Tiles Standardının Uygulanması

3D Tiles standardında, bölümler arasındaki hiyerarşik ilişki ve bölümlere ait meta veri "tileset.json" isimli dosyada tutulmaktadır. Böylece, tileset.json file dosyası istemci uygulaması tarafından ayrıştırılarak koşum zamanında hangi bölümlerin ne zaman gösterileceği belirlenir. tileset.json dosyasını üretmek için R-Tree ya da Adaptive QuadTree bölümlenmesi sonucu oluşan hiyerarşik veri yapısı bir json dosyasına kodlanır (Şekil 6).



Şekil 7. Genel sistem mimarisi

Geliştirilen yazılım Amazon Lambda Serverless sunucuları üzerinde koşturmaktadır. Kullanıcılar 3B konumsal verilerini sisteme yükleyebilmektedirler. Sunucuya yüklenen 3B konumsal veri Amazon S3 Bucket isimli veri depolama biriminde depolanmaktadır. Amazon Lambda fonksiyonları kullanılarak bu veri işlenmektedir. CityGML4J kütüphanesi ile CityGML formatındaki veri ayrıştırılmakta, earcut4j Java kütüphanesi ile de ayrıştırılan verideki 3B poligon yüzeyler üçgenlenmektedir. Daha sonra geliştirilen bölümlenme algoritmaları ile veri bölümlenerek 3D Tiles veri seti elde edilmekte, üretilen bölümler istemci uygulaması ile görüntülenebilmekte, kullanıcının yerel bilgisayarına indirilebilmektedir. İstemci uygulamasında da Cesium.js, HTML5, WebGL ve PrimeFaces gibi teknolojiler kullanılmıştır.

4. Bulgular ve Sonuçlar

Geliştirilen metodoloji ve algoritmaların testi için Trabzon ilinin 3BKM'i kullanılmıştır (Şekil 8). Bu kent veri seti ile ilgili bilgiler Tablo 1'de verilmiştir.



Şekil 8. Trabzon iline ait 3BKM'inden bir görüntü

Tablo 1. Trabzon iline ait 3BKM ile ilgili bilgiler

Veri Seti	Alanı	Obje Sayısı	Boyutu
Trabzon 3BKM	4685 km ²	56602 Adet Bina Objesi	1.05 GB

Bölümlemede kullanılan R-Tree ve Adaptive QuadTree veri yapılarının performansları oluşturulma süreleri, yeni bir obje eklenmesi, mevcut bir objenin silinmesi ve konumsal sorguların süresi göz önüne alınarak değerlendirilmiştir (Tablo 2). R-Tree (Guttman, 1984) ve Adaptive QuadTree (Pajarola vd. 2002) algoritmalarının seçilmesinin sebebi hiyerarşik ve konumsal veri yapıları olmaları, PostGIS, Oracle Spatial gibi veri tabanlarının konumsal eklentileri tarafından, 2B verilerin indekslenmesinde başarıyla yaygın olarak kullanılmalarıdır. Bu sebeple 3B konumsal verinin bölümlenmesinde de bu iki algoritma test edilmiştir. Tablo 2'de elde edilen değerler saniye birimindedir. Burada konumsal sorgu işlemi, belirli bir binaya, örneğin hastane, en yakın 100 adet diğer binaların bulunması işlemidir.

Tablo 2. Veri yapılarının karşılaştırılması

	R-Tree	Adaptive QuadTree
Oluşturulma	2513.80s	483.57s
Ekleme	1.12s	0.09s
Silme	0.90s	0.09s
Konumsal Sorgu	2.14s	4.18s

Tablo 2’deki değerler irdelendiğinde oluşturulma, ekleme, silme sürelerinde Adaptive QuadTree’nin çok daha iyi performans verdiği görülmektedir. Konumsal sorgularda ise R-Tree’nin çok saha iyi sonuç verdiği görülmektedir. Bu sonuçlara dayanarak eğer sürekli güncellenmesi gerekmeyen statik bir 3B veri varsa ve konumsal analiz ve sorgular yapılacaksa R-Tree, sürekli güncellenen bir veri seti varsa ve konumsal sorgu yapılmayacaksa veri sadece görselleştirilecekse Adaptive QuadTree veri yapısı olarak seçilmelidir.

Oluşturulan bölümler görüntülenirken, özellikle objelere yaklaşırken ya da objelerden uzaklaşırken görüntü de titremeler meydana geldiği görülmüştür. Bunun sebebi, WebGL’in coğrafi koordinatları desteklememesinden kaynaklandığı anlaşılmıştır. WebGL de noktaların koordinatları 32 bit tek duyarlıklı ondalık sayılar olarak temsil edilmektedir. Oysa koordinatlandırılmış coğrafi koordinatlar 7 basamaktan fazladır ve 32 bit tek duyarlıklı ondalık sayıların limitini aşmaktadır. Bu nedenle koordinatlarda WebGL deki veri tiplerine eşlenirken hassasiyet kaybı yaşanmaktadır. Bu durumu önlemek için her bir bölümün ECEF koordinat sistemindeki koordinatları b3dm dosyasındaki tabloda (feature table) tutularak bu değerler objelerin koordinatlarından çıkarılarak obje koordinatları lokal koordinat sistemine çevrilmiş ve basamak olarak küçültülmüştür. Böylece bu sorun giderilmiştir.

Bu çalışma kapsamında 3D Tiles test edilmiş ve irdelenmiştir. 3D Tiles büyük ölçekli verilerin web üzerinden aktarımı için tasarlanmıştır ve içerisinde barındırdığı pek çok konsepti “Bilgisayar Grafikleri” çalışma alanından almıştır. Bir CBS geliştiricisi için, bu kavram ve konseptleri anlamının ve uygulamanın kolay olmadığı ve zaman gerektirdiği görülmüştür. Dahası spesifikasyondaki bazı kavram ve tanımlamalar net değildir ve standardın uygulamasını zorlaştırmaktadır. Örneğin “geometrik hata” tanımı sözel olarak yapılmış ancak her bir bölüm için geometrik hata değerinin nasıl hesaplanacağı matematiksel olarak belirtilmemiştir.

3D Tiles standardını uygulayabilmek için her ne kadar Cesium.js, Three.js gibi JavaScript kütüphanelerinden yararlanılsa da temel seviyede WebGL bilgisini gerektirdiği görülmüştür. Bunun sebebi uygulama geliştirilirken tarayıcının WebGL ile ilgili döndürdüğü hata mesajlarının sorunların çözümü için anlaşılmasını gerektirmesidir.

3D Tiles’in kullandığı glTF kaynaklı formatların kompakt ve etkin oldukları, az yer kapladıkları ve tarayıcı tarafından hızlıca işlenebildikleri görülmüştür ancak bu formatlar ikili olmaları sebebiyle doğrudan insan tarafından okunamamakta, hata ayıklamayı ve kodlamayı zorlaştırmaktadırlar.

4. Sonuç

Bu çalışmada OGC 3D Tiles standardı, 2 farklı hiyerarşik veri yapısıyla test edilerek uygulanmıştır ve bunun sonucunda web tabanlı açık kaynak kodlu bir yazılım geliştirilmiştir. Geliştirilen bu yazılım ile kullanıcılar, herhangi bir yazılım bileşeni kurmadan web tabanlı olarak 3BKM’lerini bölümleyebilir, bölümlenen veri setini web tabanlı olarak görüntüleyebilir ve 3B bölüm setini lokal bilgisayarına indirebilir. Bölümlenme için R-Tree ve Adaptive QuadTree veri yapıları kullanılmış, veri yapıları performans bakımından kıyaslanmıştır. Yapılan kıyaslamada konumsal analizlerde R-Tree’nin, verinin oluşturulması ve güncellenmesinde de Adaptive QuadTree’nin daha iyi sonuçlar verdiği görülmüştür. 3D Tiles standardı irdelenmiş ve bazı kavramsal eksiklikler içerdiği görülmüştür.

3D Tiles standardı incelenmiş ve geliştirilen uygulama ile test edilmiştir. 3D Tiles standardının uygulanabilmesi için Bilgisayar Grafikleri ve Bilgisayar Geometrisi alanlarına ait pek çok konsept ve kavramın bilinmesi gerektiği görülmüştür.

Mevcut haliyle geliştirilen yazılım, dokuları (texture) desteklememektedir. Oysa dokular çok fazla yer kaplamakta ve bölümlenme de dikkate alınmalıdır. Şu an yürütülen mevcut çalışmalarda bölümlenme yapılırken yüzeylerin doku kaplamalarında kullanılan resimlerinde boyutunun dikkate alınması üzerine çalışılmaktadır. Ayrıca nokta bulutu verisinin de bölümlenmesi ve görüntülenmesi yine gelecekte yapılması düşünülen çalışmalar arasındadır.

Bir diğer düşünülen geliştirme de bölümlenmenin paralelleştirilerek (multi-threading), birden fazla bölümle ilgili işlemlerin aynı anda yapılması sağlanarak bölümlenme performansının artırılmasıdır.

Kaynaklar

- Chaturvedi, K., Yao, Z., & Kolbe, T. H. (2015). Web-based Exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. In *Wissenschaftlich-Technische Jahrestagung der DGPF und Workshop on Laser Scanning Applications*. 2015. (pp. 296-306). DGPF Tagungsband 24/2015.
- Gaillard, J., Vienne, A., Baume, R., Pedrinis, F., Peytavie, A., & Gesquière, G. (2015, June). Urban data visualisation in a web browser. In *Proceedings of the 20th International Conference on 3D Web Technology*. 2015. (pp. 81-88). Web3D '15.
- Gaillard, J, Peytavie, A., & Gesquiere G., (2020). Visualization and perdonalization of multi-representations city models. *International Journal of Digital Earth*, 13(5), 627-644.
- Gesquiere, G., & Manin, A. (2012). 3D Visualization of Urban Data Based on CityGML with WebGL. *International Journal of 3-D Information Modeling (IJ3DIM)*, 1(3), 1-15.
- GitHub. (2021, Aralık 10). *CesiumGS 3D-Tiles*. Retrieved from <https://github.com/CesiumGS/3d-tiles>
- Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of data*, 1984. (pp. 47-57). SIGMOD'84.
- Koukofikis, A., Coors, V., & Gutbell, R. (2018). Interoperable visualization of 3d city models using OGC's standard 3D portrayal service. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(4), 113-118.
- Kramer, M, & Gutbell R., (2015). A case study on 3D geospatial applications in the Web using state-of-the-art WebGL frameworks. In *Proceedings of the 20th International Conference on 3D Web Technology*, 2015. (pp. 189-197). Web3D '15.
- Lu, M., Wang, X., Liu, X., Chen, M., Bi, S., Zhang, Y., and Lao, T. (2020). Web-based real-time visualization of large-scale weather radar data using 3D tiles. *Transactions in GIS*, 25(1), 25-43.
- Pajarola, R., Antonijuan, M., & Lario, R. (2002). Quadtree based triangulated irregular networks. In *IEEE Visualization, 2002, VIS 2002*. (pp. 395-402). IEEE.
- Prandi, F., De Amicis, R., Piffer, S., Soave, M., Cadzow, S., Boix, E. G., & D'Hondt, E. (2013). Using CityGML to deploy smart-city services for urban ecosystems. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W1(2013), 87-92.
- Xu, Z., Zhang, L., Li, H., Lin, Y. H., & Yin, S. (2020). Combining IFC and 3D tiles to create 3D visualization for building information modeling. *Automation in Construction*, 109, 102995, doi: 10.1016/j.autcon.2019.102995.