

**İNANDIRICI KALABALIK BENZETİMİNDE BİREY SAYISI
VE HAREKET ÇEŞİTLİLİĞİ ARAŞTIRMASI**

**INVESTIGATION OF AGENT NUMBER AND MOVEMENT
VARIATIONS IN CONVINCING CROWD SIMULATION**

ÖNER BARUT

PROF. DR. EBRU AKÇAPINAR SEZER

Tez Danışmanı

DR. MURAT HACIÖMEROĞLU

İkinci Tez Danışmanı

Hacettepe Üniversitesi

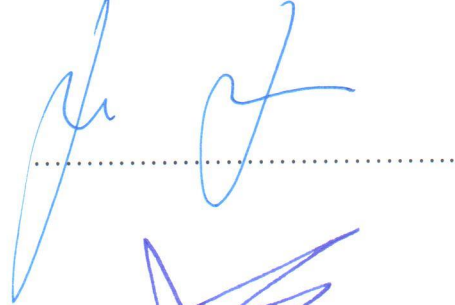
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü

DOKTORA TEZİ olarak hazırlanmıştır.

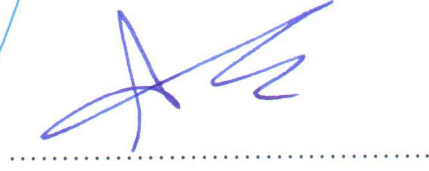
2017

ÖNER BARUT'un hazırladığı "İnandırıcı Kalabalık Benzetiminde Birey Sayısı ve Hareket Çeşitliliği Araştırması" adlı bu çalışma aşağıdaki jüri üyeleri tarafından BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda DOKTORA TEZİ olarak kabul edilmiştir.

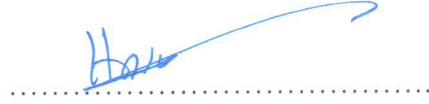
Prof. Dr. Hayri SEVER
Başkan



Prof. Dr. Ebru AKÇAPINAR SEZER
Danışman



Prof. Dr. Haşmet GÜRÇAY
Üye



Doç. Dr. Hacer KARACAN
Üye



Yrd. Doç. Dr. Mehmet Serdar GÜZEL
Üye



Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafında DOKTORA TEZİ olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesi'ne verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**

(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

- Tezimin/Raporumun 26 / 09 / 2018 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**

(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

- Tezimin/Raporumun tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

- Serbest Seçenek/Yazarın Seçimi**

26 / 09 / 2017



ÖNER BARUT

Aileme...

ETİK

Hacettepe Üniversite Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili esere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

13 / 09 / 2017

ÖNER BARUT

ÖZET

İNANDIRICI KALABALIK BENZETİMİNDE BİREY SAYISI VE HAREKET ÇEŞİTLİLİĞİ ARAŞTIRMASI

Öner BARUT

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Ebru AKÇAPINAR SEZER

İkinci Tez Danışmanı: Dr. Murat HACİÖMEROĞLU

Eylül 2017, 193 sayfa

Günümüzde eğlence sektörünün önemli bir bölümünü oluşturan bilgisayar oyunları ve filmler kalabalık benzetimlerini yoğun bir biçimde kullanan başlıca uygulama alanları arasında yer almaktadır. Bu gibi eğlence uygulamalarının çoğunda kalabalıklar var olan sahne içerisinde bir arka plan oluşturulması amacıyla kullanılmaktadır. Ambiyans kalabalığı olarak isimlendirilen bu benzetimlerde kalabalığı oluşturan bireylerin mümkün olan en yüksek kalite ile görselleştirilmesine ihtiyaç olmadığı gibi bireylerin gezinimini sağlamak adına yüksek maliyetli karmaşık işlemler yapılmasına da gerek yoktur. Etkileşim içermeyen ambiyans kalabalıklarının gerçek zamanlı olarak oluşturulması ve mümkün olan en düşük gezinim maliyetiyle benzetimlerinin gerçekleştirilmesi amacıyla bu tez çalışması kapsamında 2 boyutlu benzetim alanları üzerinde inandırıcılıktan ödün vermeden maksimum yoğunluğa ve minimum hareket çeşitliliğine sahip kalabalık benzetimleri oluşturulması için yeni bir yaklaşım önerilmiştir. Buna göre, kalabalığı oluşturan bireylerin gezinim hareketlerinin yüksek işlem gücü ve işletim süresi gerektiren çarpışma tespiti ve kaçınma manevrası yapma üzerine kurulu gezinim modelleri ile sağlanması yerine, bireylerin kendilerine başlangıç noktasından bitiş noktasına kadar tanımlanan ve çarpışmasız

olduđu garanti edilen yörüngelerde gezinmeleri sağlanmıştır. Benzetime katılacak olan bireylere, oluşturulmaları esnasında atanacak bu ön tanımlı yörüngelerin belirlenmesi için üç farklı yönlendirmesiz gezinim yöntemi geliştirilmiştir. Bu yöntemler alanın en güncel birey tabanlı gezinim tekniklerinden biriyle karşılaştırılmış ve her bir yöntemin de kıyaslandıkları metottan çok daha düşük gezinim maliyetlerine sahip olduğu görülmüştür. Buna ek olarak, geliştirilen yönlendirmesiz gezinim yaklaşımının çok daha tutarlı ve pürüzsüz yörüngeler izleyerek hareket eden bireyler ve gerçekçilik algısı daha yüksek gezinim davranışları sergileyen kalabalıklar oluşturabileceđi gösterilmiştir.

Anahtar Kelimeler: kalabalık benzetimi, ambiyans kalabalığı, yönlendirmesiz gezinim, konum-zamansal yörünge engeli, doğrusal yörünge parçası, sabit hızlı yörünge parçası

ABSTRACT

INVESTIGATION OF AGENT NUMBER AND MOVEMENT VARIATIONS IN CONVINCING CROWD SIMULATION

Öner BARUT

Doctor of Philosophy, Department of Computer Engineering

Supervisor: Prof. Dr. Ebru AKÇAPINAR SEZER

Co-Supervisor: Dr. Murat HACIÖMEROĞLU

September 2017, 193 pages

Nowadays, computer games and movies, which constitute a significant part of the entertainment industry, are among the main application areas that use crowd simulations extensively. In the majority of such entertainment applications, crowds are used to create a background within the existing scene. In these simulations named as ambient crowds, there is no need to visualize the individuals forming the crowd with the highest possible quality or to perform costly complex operations to navigate these individuals. Within the scope of this thesis study, a new approach has been proposed to create crowd simulations having maximum density and minimum movement variety on 2D simulation areas without compromising plausibility for the purpose of creating non-interactive ambient crowds in real-time and simulating these crowds with the lowest possible navigation cost. According to this, instead of navigating crowd members with the navigation models based on collision detection and avoidance maneuvers requiring a high computational power and operation time, individuals are navigated on trajectories that are defined from their initial positions to their goal positions and are guaranteed to be collision-free. Three different steering-free navigation methods have been developed for the creation of the trajectories that will be

assigned to the individuals who will participate in the simulation. These methods have been compared with one of the state-of-the-art agent-based navigation techniques existing in the literature and each of these methods have been shown to have much lower navigation costs than the technique they compared. In addition, it has been shown that the steering-free navigation approach that has been developed can create individuals that move on more consistent and smooth trajectories and crowds that exhibit navigation behaviors having a higher level of perceived realism.

Keywords: crowd simulation, ambient crowd, steering-free navigation, spatio-temporal trajectory obstacle, linear trajectory segment, constant-speed trajectory segment

TEŞEKKÜR

Tez çalışmalarım süresince engin bilgi birikiminin yanı sıra gerek akademik gerekse hayata dair tecrübelerini benimle paylaşarak bu süreçte bana hoşgörü ve sabırla destek olmanın ötesinde bilhassa doktora sürecimin en zor dönemlerinde ışığıyla beni ve yolumu aydınlatan danışmanım Sayın Prof. Dr. Ebru AKÇAPINAR SEZER'e,

Lisansüstü öğrenim sürecimin ilk dönemlerinden itibaren bilgisayar grafiği gibi bilgisayar bilimlerinin en zorlu alanlarından birinde büyük bir özveri ile her daim bana yol gösteren eş danışmanım Sayın Murat HACIÖMEROĞLU'na,

Lisans öğrenimim esnasında ve sonrasında bana akademik kariyer yapma fikrini aşlamakla birlikte beni bu yolda yürümek konusunda cesaretlendirerek akademisyenliğe başlamama vesile olan Sayın Prof. Dr. Hayri SEVER'e,

Çok değerli yorum ve önerileriyle tez çalışmalarım değerli katkılarını sunan tez savunma sınavı jüri üyelerim Sayın Prof. Dr. Haşmet GÜRÇAY'a, Sayın Doç. Dr. Hacer KARACAN'a, Sayın Yrd. Doç. Dr. Mehmet Serdar GÜZEL'e ve bir dönem tez izleme komitesi üyeleri arasında yer alan Yrd. Doç. Dr. Erhan MENGÜŞOĞLU'na,

Hem lisans hem de lisansüstü öğrenimimi tamamladığım ve bu on yılı aşkın süre sonrasında kendimi de bir parçası olarak görmekten çok büyük gurur ve mutluluk duyduğum Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nün tüm akademik ve idari çalışanlarına,

Lisans öğrenimime başladığım ilk andan lisansüstü öğrenimimi tamamladığım son ana kadar fikir alışverişinde bulunduğum sınıf arkadaşım, mesai arkadaşım ve en önemlisi yol arkadaşım Sayın Cumhur Yiğit ÖZCAN başta olmak üzere maddi ve manevi desteğini esirgemeyen tüm diğer arkadaşlarıma,

Çocukluğumdan beri en yakın destekçim olmakla kalmayıp yaşamımın devamında da bunu en mükemmel şekilde sürdüreceğine hiç şüphe duymadığım kardeşime,

Hayata gözlerimi açtığım ilk andan itibaren hiçbir fedakârlıktan kaçınmayarak sevgi ve ilgileriyle bugünümün ve yarınlarımla baş mimarları olan anneme ve babama teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
ÇİZELGELER	viii
ŞEKİLLER	ix
SİMGELER VE KISALTMALAR	xix
1. GİRİŞ.....	1
1.1. Tez Çalışmasının Motivasyonu	1
1.2. Tez Çalışmasının Temel Katkıları	5
1.3. Tez Metninin Organizasyonu	8
2. ALAN BİLGİSİ VE ALANYAZIN ÖZETİ	10
2.1. Kalabalık ve Kalabalık Benzetimi	10
2.2. Kalabalık Benzetiminin Kullanım Alanları	11
2.3. Kalabalık Benzetiminin Bileşenleri.....	14
2.3.1. Kalabalık Benzetiminde Davranış Modelleme.....	14
2.3.2. Kalabalık Benzetiminde Görselleştirme	29
2.4. Kalabalık Benzetiminin Kısıtlamaları	29
2.5. Kalabalık Benzetiminde Detay Seviyesi	31
2.5.1. Görselleştirme Detay Seviyesi.....	32
2.5.2. Davranış Detay Seviyesi.....	38
2.6. Kalabalık Benzetimlerinde İnanırcılık	42
3. ANA İŞLEM BİRİMİ TABANLI TEK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ.....	48
3.1. Çarpışmasız Sabit Hız Vektörleri Belirlenmesi.....	48
3.1.1. Rastgele Başlangıç Pozisyonu ve Hız Vektörü Seçilmesi.....	48
3.1.2. Çarpışmalara Neden Olabilecek Bireylerin Kümesinin Belirlenmesi	49
3.1.3. Kesin Çarpışma Testlerinin Gerçekleştirilmesi.....	52
3.2. Bireylerin Benzetim Alanı Üzerinde Hareket Ettirilmesi.....	53
4. GRAFİK İŞLEM BİRİMİ TABANLI TEK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ.....	56

4.1. TYPÇY Bulma Problemi.....	56
4.2. Probleme Getirilen Çözüm	58
4.3. Yeni Birey Oluşturma Sürecinin Detayları	59
4.3.1. Bireylerin Konum-Zamansal Görselleştirilmesi.....	60
4.3.2. Olası Çözümlerin Analiz Edilmesi	68
4.4. Benzetimde Yer Alan Bireylerin Hareket Ettirilmesi.....	76
5. ÇOK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ	77
5.1. Yeni Birey Oluşturulması.....	77
5.1.1. Kameranın Ayarlanması	79
5.1.2. Bireylerin Konum-Zamansal Görselleştirilmesi.....	80
5.1.3. Olası Çözümlerin Analiz Edilmesi	84
5.2. Bireylerin Hareket Ettirilmesi	92
6. DENEYLER VE SONUÇLARI.....	95
6.1. TYPÇY Yöntemlerinin Değerlendirilmesi	96
6.1.1. Grup Merkezi Seçim Yöntemlerinin İrdelenmesi	96
6.1.2. Birey Oluşturma Başarımının Karşılaştırılması	99
6.2. ÇYPÇY Yönteminin Değerlendirilmesi.....	101
6.2.1. Birey Oluşturma Başarımının Karşılaştırılması	101
6.2.2. Çözüm Seçim Yöntemlerinin İrdelenmesi	118
6.3. Geliştirilen Gezinim Yöntemlerinin Maliyet Analizi.....	127
6.3.1. Birey Oluşturma Maliyetlerinin Karşılaştırılması	128
6.3.2. Birey Gezinimi Maliyetlerinin Karşılaştırılması	142
6.3.3. Bellek Maliyetlerinin Karşılaştırılması	152
6.4. Sabit Engeller İçeren Benzetimlerdeki Birey Oluşturma Başarımının İncelenmesi ..	154
6.4.1. Birey Yarıçapının 0.5 m Değeri için Sonuçlar	157
6.4.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar	159
6.5. Geliştirilen Gezinim Yöntemlerinin Yörünge Kalitesinin Değerlendirilmesi.....	162
7. TARTIŞMA VE ÖNERİLER.....	169
8. SONUÇ.....	181
KAYNAKLAR.....	183
ÖZGEÇMİŞ.....	192

ÇİZELGELER

Sayfa

Çizelge 6.1 Grup merkezi seçim yöntemi ve benzetim alanı büyüklüğünün etki büyüklükleri	98
Çizelge 6.2 Gezinim yöntemi ve benzetim alanı büyüklüğünün etki büyüklükleri	101
Çizelge 6.3 GİB tabanlı TYPÇY ve yalnızca TYPÇY'ler kullanan ÇYPÇY yöntemleri ile birey oluşturma için ayrılan süre parametresinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)	106
Çizelge 6.4 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)	111
Çizelge 6.5 GİB tabanlı TYPÇY ve yalnızca TYPÇY'ler kullanan ÇYPÇY yöntemleri ile birey oluşturma için ayrılan süre parametresinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)	112
Çizelge 6.6 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)	118
Çizelge 6.7 ÇYPÇY yönteminde kullanılan çözüm seçim yöntemlerinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$) ..	120
Çizelge 6.8 ÇYPÇY yönteminde kullanılan çözüm seçim yöntemlerinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$) ..	124
Çizelge 6.9 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey oluşturma süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)	135
Çizelge 6.10 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey oluşturma süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)	141
Çizelge 6.11 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey başına gezinim süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)	147
Çizelge 6.12 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey başına gezinim süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)	152

ŞEKİLLER

Sayfa

Şekil 1.1 Örnek bir kalabalık benzetimi sahnesi [1].....	1
Şekil 1.2 Kalabalık parçaları yöntemiyle oluşturulmuş bir kalabalık benzetiminin örnek sahne görüntüsü [6]	3
Şekil 1.3 Gerçekleştirilen psikolojik bir çalışmada [9] doğrusal çarpışmasız yörüngeler üzerinde sabit hızla hareket eden dairesel nesnelere	5
Şekil 2.1 Alan bilgisi ve alanyazın özetinde ele alınan başlıkların hiyerarşik yapısı.....	10
Şekil 2.2 Gerçek bir kalabalık fotoğrafı ve bu kalabalığın benzetimi yapılan sanal gerçekleştirimi [17].....	11
Şekil 2.3 Acil durum boşaltma senaryolarında kalabalık benzetimi kullanımı [24]	12
Şekil 2.4 Trafik sinyal ve işaretleri bulunan dört yönlü bir kavşaktaki araç trafiğinin benzetimi [35].....	13
Şekil 2.5 Büyük bir göçmen çekirge sürüsünün küçük bir köyü istilasının benzetimi [42]	13
Şekil 2.6 Bireylerin otonom varlıklar olarak modellenmesi için kullanılan geleneksel yaklaşım [48]	15
Şekil 2.7 Kalabalık benzetimi yöntemlerinin sırasıyla büyük ölçekli (makroskobik), orta ölçekli (mezoskopik) ve küçük ölçekli (mikroskobik) olmak üzere üç ana başlık altına gruplanması [51].....	17
Şekil 2.8 Treuille ve arkadaşları [57] tarafından geliştirilen büyük ölçekli modelin genel algoritması	19
Şekil 2.9 Reynolds [58] tarafından geliştirilen kural tabanlı yöntem ile oluşturulmuş hayvan sürüsü benzetimi	20
Şekil 2.10 Dar bir geçitte yer alan kapıyı sırayla iki farklı yöne hareket eden bireylerin kullanması [61].....	21
Şekil 2.11 Disk şeklinde ifade edilen A bireyinin yine disk şeklindeki B bireyinden kaynaklı hız vektörü engeli [64].....	22
Şekil 2.12 (Sol) Hız vektörü engeli yöntemi ile tam manevra yapılması sonucu oluşan salınımlar, (Sağ) Kısmi manevra yapılması ile salınım probleminin önlenmesi [64]	23
Şekil 2.13 Kalabalık parçaları yönteminin yapı taşları olan parçalardan birinin farklı görünüşleri [7].....	25
Şekil 2.14 Benzetim alanının farklı bölgelerinde ihtiyaca göre değişik modellerin kullanılması [70].....	26

Şekil 2.15 Narain ve arkadaşları [72] tarafından geliştirilen karma modelin genel sistem yapısı.....	27
Şekil 2.16 Tissera ve arkadaşları [76] tarafından geliştirilen modelin katman yapısı.....	28
Şekil 2.17 Farklı kalabalık modelleri ve uygulamalarının zaman ölçeği ve büyüklük açısından sınıflandırması [16]	31
Şekil 2.18 Poligon sayısı farklı oranlarda azaltılmış 3 boyutlu model [81]	33
Şekil 2.19 İlerlemeli poligon örgüleri yönteminde poligon artırma (<i>vsplit</i>) ve poligon azaltma (<i>ecol</i>) işlemleri [83]	33
Şekil 2.20 Poligon sayısı kelebek yöntemiyle artırılan 3 boyutlu model [89]	34
Şekil 2.21 Her bir animasyon pozunu için farklı açılardan resim çekilmesi [92]	35
Şekil 2.22 Bir animasyon pozunu için farklı açılardan çekilerek kaydedilmiş resimler [92] .	35
Şekil 2.23 Animasyon pozuna ve kamera açısına uygun olarak çekilmiş resmin sürekli kameraya bakan dikdörtgen tabelaya kaplanması [93].....	36
Şekil 2.24 Modelin farklı uzuvlarına ait impostor resimlerinin farklı animasyon pozları için deforme edilerek kullanılması [95].....	37
Şekil 2.25 (Sol) Çekilen resimlerin düz bir tabelaya kaplandığı 2 boyutlu impostor kullanmak yerine (Sağ) çekilen resimlerin modelin daha az poligon sayısına sahip düşük çözünürlüklü bir haline kaplanması [96]	37
Şekil 2.26 Bireylerin hareketli uzuvlarının her birini bir sınırlayıcı kutu içerisine alarak bu kutuların her yüzüne ilgili uzvun o açıdan çekilmiş renk, normal ve derinlik bilgisi içeren resimlerinin kaplanması [97]	38
Şekil 2.27 Hareketli eklem sayısının azaltılarak animasyon kalitesinin düşürülmesi [2] ...	39
Şekil 2.28 Hareketli eklem sayısının değiştirilerek iskelet yapısının basitleştirilmesi [99]	40
Şekil 2.29 Kameranın görüş alanının farklı bölümlerine uygulanan üç farklı detay seviyesi [4]	41
Şekil 2.30 Kalabalık benzetiminde yer alan bireylerin yarıçaplarının detay seviyesine göre ayarlanması [3]	41
Şekil 2.31 Üç farklı detay seviyesindeki bireyler için kullanılan sınırlayıcı küreler [100].	42
Şekil 2.32 Giysilerin kumaş benzetiminde kullanılan farklı sertliklerdeki materyaller [105]	45
Şekil 2.33 Grup halindeki bireylerin diyaloglarını gösteren örnek bir sahne [108].....	46
Şekil 2.34 Bir karakterin diğer bir karakteri eliyle ittiği sanal bir sahne [115].....	47
Şekil 2.35 En soyuttan en gerçekçi olana doğru sıralanmış farklı görselleştirme biçimleri [116]	47

Şekil 3.1 Benzetim alanı üzerinde (Sol) düzensiz yoğunluk dağılımı ve (Sağ) düzenli yoğunluk dağılımını örnekleyen senaryolar	49
Şekil 3.2 Bireylerin benzetim alanı üzerinde kapladıkları dairesel alanın temas edeceği tüm hücrelere kimlik numaralarının yanı sıra benzetimin başlangıcına göreli ilk temas anı ve son temas anının kaydedilmesi	50
Şekil 3.3 Bireylerin 2 boyutlu benzetim alanı üzerinde r yarıçaplı diskler ile ifade edilmesi	51
Şekil 3.4 Yeni birey için oluşturulan aday yolun, 2 boyutlu benzetim alanı üzerinde temas edeceği hücrelerin belirlenmesi için (Sol) eksik çözümler üretebilecek bireyin yörüngesine karşılık gelen doğru parçası yerine (Sağ) bireyin yörüngesine paralel ve temsil edildiği diske teğet iki farklı doğru parçası ile kesişimlerin hesaplanması	51
Şekil 4.1 (Sol) Bireylerin pozisyonlarının hız vektörlerine bağlı olarak konum-zamansal uzay içerisinde izleyeceği yörünge, (Sağ) Bireylerin 2 boyutlu benzetim alanı üzerinde disk biçiminde ifade edilmesinden dolayı konum-zamansal uzay içerisinde kaplayacağı boru biçimindeki hacim	57
Şekil 4.2 Benzetime dâhil olacak yeni bir birey oluşturmak için o anda S uzayında var olan yoğun boru (koyu gri) kümesi içerisinde uygun bir boşluk bulunarak diğer hiçbir boruyla kesişmeyecek biçimde yeni bireyin konum-zamansal yörüngesine karşılık gelmek üzere yeni bir boru (yeşil) yerleştirilmesi	58
Şekil 4.3 Çarpışmasız hız vektörü belirleme süreci (a) Benzetim alanının kenarlarından biri üzerinde rastgele bir başlangıç pozisyonu (yeşil nokta) seçilmesi (b) Kameranın ayarlanması ve var olan bireylerin konum-zamansal borularının görselleştirilmesi (c) Oluşturulan konum-zamansal sahnenin kameranın bakış açısından görünümü (d) Sahnenin kameranın bakış açısından elde edilen siyah beyaz resmi (e) Ön tanımlı hız aralığı içerisinde sonuçlar üreten çözümlerin (mavi olmayan pikseller) belirlenmesi (f) Beyaz bir piksel seçerek çarpışmasız sabit hız vektörünün (bitiş pozisyonu ve yürüme hızı) belirlenmesi.....	60
Şekil 4.4 Benzetim alanının kuşbakışı ortografik iz düşüm resimlerinin üst üste yığılmasıyla oluşan silindirik borulardan meydana gelen 3 boyutlu sahne.....	61
Şekil 4.5 Perspektif kamerada görüş hacmi [119].....	63
Şekil 4.6 Perspektif kamerada görüş hacminin büyüklüğünü belirleyen faktörler [120]....	63
Şekil 4.7 Perspektif kamerada görüntü oluşumu [121]	64
Şekil 4.8 Benzetim alanının boyutları ve kameranın görüş hacminin benzetim alanı üzerine iz düşümü.....	65
Şekil 4.9 Kameranın pozisyonunun ve görüş hacminin ayarlanması	66

Şekil 4.10 Yeni bireyin dairesel çapının önceden hesaba katılarak sadece bir doğru ile ifade edilmesi.....	68
Şekil 4.11 Seçilen beyaz bir piksele göre bitiş pozisyonu ve yürüme hızı değerini belirlenmesi	69
Şekil 4.12 Beyaz piksel gruplarının merkezlerini bulmak için gerçekleştirilen ilk geçişin algoritması	72
Şekil 4.13 Mavi renk ile gösterilen beyaz doku hücresinin işlendiği varsayıldığında sadece kendisinden daha küçük doku hücresi kimlik numarasına sahip olabilecek turuncu renk ile gösterilmiş doku hücrelerinin kontrol edilmesi	73
Şekil 4.14 Beyaz piksel gruplarının merkezlerini bulmak için gerçekleştirilen ikinci geçişin algoritması	75
Şekil 5.1 Tek bir doğrusal ve sabit hızlı yörünge parçasına sahip olan bir bireyin 3 boyutlu konum-zaman uzayında kapladığı silindirik hacim.....	78
Şekil 5.2 Konum-zamansal görselleştirme öncesi perspektif kameranın ayarlanması.....	80
Şekil 5.3 Her bir yolun sahip olduğu her bir yörünge parçası için oluşturulan konum-zamansal hacimler	82
Şekil 5.4 (Sol) Kameranın bakış açısından silindirik borulardan oluşan konum-zaman uzayının görünümü, (Orta) Kameranın bakış açısından elde edilmiş siyah beyaz iz düşüm resmi, (Sağ) Derinlik belleği içeriğinin doğrusallaştırılmış ve normalleştirilmiş derinlik değerlerine göre oluşturulmuş monokrom resmi.....	82
Şekil 5.5 (Üst-Sol) Var olan bireylerin borularının orijinal yarıçapları ile görselleştirilmesi (Üst-Sağ) durumunda yeni birey oluşturmak için gereken yeterli genişlikte bitişik beyaz piksel kümesi bulmak gerekir. (Alt-Sol) Var olan bireylerin borularının orijinal yarıçaplarının iki katı ile görselleştirilmesi sonucu yeni bireyi sadece bir doğru ile ifade etmek (Alt-Sağ) ve tek bir beyaz pikselin çarpışmasız bir çözüm olduğunu garanti etmek mümkündür.....	83
Şekil 5.6 Siyah pikselin derinlik değeri ve pay katsayısı kullanılarak yörünge parçasının bitim noktasının belirlenmesi	86
Şekil 5.7 Birden çok yörünge parçasından oluşan ve bu yörünge parçaları arasındaki geçişlerde doğrultu ve hız değişiklikleri barındıran örnek bir çarpışmasız yolun adım adım oluşturulması	87
Şekil 5.8 Çözüm olarak kullanılacak pikseli seçmek için gerçekleştirilen ilk geçişin algoritması	88

Şekil 5.9 Çözüm olarak kullanılacak pikseli seçmek için gerçekleştirilen ikinci geçişin algoritması	90
Şekil 6.1 Grup merkezi seçim yöntemlerinin 128 <i>m</i> kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları	97
Şekil 6.2 Grup merkezi seçim yöntemlerinin 256 <i>m</i> kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları	97
Şekil 6.3 Gezinim yöntemlerinin 128 <i>m</i> kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları.....	100
Şekil 6.4 Gezinim yöntemlerinin 256 <i>m</i> kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları.....	100
Şekil 6.5 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	106
Şekil 6.6 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	106
Şekil 6.7 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	107
Şekil 6.8 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	107
Şekil 6.9 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$).....	108
Şekil 6.10 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$).....	108
Şekil 6.11 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$).....	109
Şekil 6.12 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$).....	109

Şekil 6.13 (Sol) GİB tabanlı TYPÇY ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin örnek benzetim alanı görünüşleri.....	110
Şekil 6.14 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	113
Şekil 6.15 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	113
Şekil 6.16 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	114
Şekil 6.17 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	114
Şekil 6.18 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$).....	115
Şekil 6.19 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$).....	115
Şekil 6.20 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$).....	116
Şekil 6.21 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$).....	116
Şekil 6.22 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$).....	121
Şekil 6.23 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	121
Şekil 6.24 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	121

Şekil 6.25 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)	122
Şekil 6.26 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$).....	122
Şekil 6.27 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)	122
Şekil 6.28 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)	123
Şekil 6.29 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)	123
Şekil 6.30 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$).....	125
Şekil 6.31 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	125
Şekil 6.32 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	125
Şekil 6.33 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)	126
Şekil 6.34 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$).....	126
Şekil 6.35 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)	126
Şekil 6.36 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)	127

Şekil 6.37 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)	127
Şekil 6.38 RVO ve AİB tabanlı TYPÇY yöntemleriyle gerçekleştirilen benzetimler için ortalama birey oluşturma süreleri ($r = 0.5 m$)	130
Şekil 6.39 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)	130
Şekil 6.40 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)	131
Şekil 6.41 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)	131
Şekil 6.42 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)	131
Şekil 6.43 GİB tabanlı TYPÇY yönteminde başarılı ve başarısız birey oluşturma denemeleri için harcanan ortalama süreler ($r = 0.5 m$)	133
Şekil 6.44 GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere yörünge parçası belirleme sürecinin farklı alt işlemleri tarafından harcanan süreler ($r = 0.5 m$)	134
Şekil 6.45 RVO ve AİB tabanlı TYPÇY yöntemleriyle gerçekleştirilen benzetimler için ortalama birey oluşturma süreleri ($r = 0.3 m$)	136
Şekil 6.46 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)	136
Şekil 6.47 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)	137
Şekil 6.48 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)	137

Şekil 6.49 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)	137
Şekil 6.50 GİB tabanlı TYPÇY yönteminde başarılı ve başarısız birey oluşturma denemeleri için harcanan ortalama süreler ($r = 0.3 m$)	139
Şekil 6.51 GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere yörünge parçası belirleme sürecinin farklı alt işlemleri tarafından harcanan süreler ($r = 0.3 m$)	140
Şekil 6.52 RVO yöntemiyle gerçekleştirilen benzetim için birey başına ortalama gezinim süresi ($r = 0.5 m$)	144
Şekil 6.53 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)	145
Şekil 6.54 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)	145
Şekil 6.55 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)	145
Şekil 6.56 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)	146
Şekil 6.57 RVO yöntemiyle gerçekleştirilen benzetim için birey başına ortalama gezinim süresi ($r = 0.3 m$)	149
Şekil 6.58 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)	149
Şekil 6.59 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)	150
Şekil 6.60 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)	150

Şekil 6.61 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)	150
Şekil 6.62 (Sol) Bir kenarı 128 <i>m</i> olan kare biçiminde bir benzetim alanı, (Sağ) kısa kenarı 128 <i>m</i> ve uzun kenarı 192 <i>m</i> olan dikdörtgen biçiminde bir benzetim alanı ve bu alanlar (gri) üzerinde yer alan daire ve kare biçimindeki sabit engeller (siyah)	156
Şekil 6.63 Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.5 m$)	158
Şekil 6.64 Dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.5 m$)	158
Şekil 6.65 Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.3 m$)	160
Şekil 6.66 Dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.3 m$)	161
Şekil 6.67 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 0. saniyesinde yakalanan anlık kuşbakışı görüntüleri	165
Şekil 6.68 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 10. saniyesinde yakalanan anlık kuşbakışı görüntüleri	166
Şekil 6.69 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 20. saniyesinde yakalanan anlık kuşbakışı görüntüleri	166
Şekil 6.70 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 30. saniyesinde yakalanan anlık kuşbakışı görüntüleri	167
Şekil 6.71 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 40. saniyesinde yakalanan anlık kuşbakışı görüntüleri	167
Şekil 6.72 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 50. saniyesinde yakalanan anlık kuşbakışı görüntüleri	168
Şekil 6.73 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 60. saniyesinde yakalanan anlık kuşbakışı görüntüleri	168

SİMGELER VE KISALTMALAR

Simgeler

c_x	İz Düşüm Resmindeki Piksellerin Resmin Köşesine Göreli Yatay Koordinatı
c_y	İz Düşüm Resmindeki Piksellerin Resmin Köşesine Göreli Dikey Koordinatı
d_{diff}	Bireyin Bitiş Noktasına Ulaştıktan Sonra Aldığı Mesafe
d_{env}	Düzlemsel Benzetim Alanının Derinliği
d_{max}	Bireylerin Maksimum Seyahat Mesafesi
d_{min}	Bireylerin Minimum Seyahat Mesafesi
k	İz Düşüm Resminin Ölçeklendirme Katsayısı
$p_{current}$	Bireyin Güncel Pozisyonu
P_g	Bireyin Konum-Zamansal Uzaydaki Bitiş Pozisyonu
P_{goal}	Bireyin Benzetim Alanı Düzlemi Üzerindeki Bitiş Pozisyonu
P_i	Bireyin Konum-Zamansal Uzaydaki Başlangıç Pozisyonu
P_{init}	Bireyin Benzetim Alanı Düzlemi Üzerindeki Başlangıç Pozisyonu
p_{new}	Bireyin Benzetim Alanı Düzlemi Üzerindeki Yeni Hesaplanan Pozisyonu
r	Bireylerin 2 Boyutlu Benzetim Alanı Üzerinde Kapladığı Alanın Yarıçapı
S	Konum-Zamansal Uzay
s_{max}	Bireylerin Maksimum Yürüme Hızı
s_{min}	Bireylerin Minimum Yürüme Hızı
T	Konum-Zamansal Uzayda Zaman Eksen
t_{diff}	Bireyin Bitiş Noktasına Ulaşmasından Sonra Geçen Süre
t_{int}	Bireylerin Seyahat Süresi Aralığı
t_{max}	Bireylerin Maksimum Seyahat Süresi
t_{min}	Bireylerin Minimum Seyahat Süresi
U	Konum-Zamansal Uzayda Yatay Eksen
v	Bireyin Hız Vektörü
V	Konum-Zamansal Uzayda Derinlik Eksen
w_{env}	Düzlemsel Benzetim Alanının Genişliği
X	3 Boyutlu Uzayda Yatay Eksen

Y	3 Boyutlu Uzayda Dikey Eksen
y_{coord}	Doku Resmindeki Doku Hücrelerinin Resmin Köşesine Göreli Dikey Koordinatı
Z	3 Boyutlu Uzayda Derinlik Ekseni
α	Bireyin Güncel Pozisyonundan Bitiş Pozisyonuna Uzanan Vektör ile Hız Vektörü Arasındaki Açık
Δt	Benzetimin Önceki Zaman Adımından Beri Geçen Süre
ϵ	Çok Küçük Ölçeklendirme Ek Katsayısı

Kısaltmalar

AİB	Ana İşlem Birimi
BBE	Bağlantılı Bileşen Etiketleme
ÇYPCY	Çok Yörünge Parçalı Çarpışmasız Yol
DS	Detay Seviyesi
EYYN	En Yakın Yaklaşma Noktası
GİB	Grafik İşlem Birimi
GLSL	OpenGL Shading Language
GPGPU	General Purpose Computing on Graphics Processing Unit
KYM	Kırmızı, Yeşil, Mavi
KZYE	Konum-Zamansal Yörünge Engeli
RKAN	Resim Karesi Arabellek Nesnesi
RVO	Reciprocal Velocity Obstacles
SBRK	Saniye Başına Resim Karesi
TYPÇY	Tek Yörünge Parçalı Çarpışmasız Yol

1. GİRİŞ

1.1. Tez Çalışmasının Motivasyonu

Oldukça geniş bir kullanım alanına sahip olan kalabalık benzetimleri psikoloji, sosyoloji, acil durum planlama, ulaşım planlama gibi bilimsel uygulamaların yanı sıra animasyon filmler, bilgisayar oyunları, sanal gerçeklik uygulamaları ve mimari tasarım gibi artistik ve estetik amaçlar için de hizmet etmektedir (Şekil 1.1). Tüm bu alanlarda çalışmakta olan araştırmacılar, ortak problemleri olan sürekli artan sayıda bireyin ihtiyaç duyulan görsel ve davranışsal gerçekçilik seviyesinde benzetiminin gerçekleştirilmesi için çaba göstermektedir.



Şekil 1.1 Örnek bir kalabalık benzetimi sahnesi [1]

Çoğu benzetim sistemlerinde olduğu gibi kalabalık benzetimlerinde de gerçekte yüzde yüz örtüşen bir model oluşturulması ve bunun ispatlanması mümkün değildir. Bu husus gerçek hayat ile benzetim arasındaki benzerliğin öznel değerlendirilebilir olmasından kaynaklanır. Aynı zamanda gerçekte yüzde yüz örtüşen bir kalabalık benzetiminin gerçek zamanlı gerçekleştirimi mümkün değildir. Çünkü kalabalık benzetimlerinde gerçek dünyaya yakınsamak adına işlem maliyeti yüksek çok fazla hesaplama yapılmaktadır. Bir başka deyişle, benzetim başarımı ile işlem karmaşıklığı arasında güçlü ve doğru orantılı bir ilişki tespit edilebilir. Bu noktada ortaya koyulan çözümler benzetimin kullanılacağı sektörün

ihtiyalarına gre Őekillenmektedir. Gerek zamanda alıŐma zorunluluĐunun bulunmadıĐı kalabalık uygulamaları gereĐe olabildiĐince yakın benzetimler sunabilirken, gerek zaman kısıtının kaınılmaz olduĐu kalabalıklarda ise daha ok kullanıcılar tarafından gerekmiŐ gibi algılanabilen yani inandırıcı kalabalık benzetimleri tercih edilmektedir.

Bilimsel amalarla kullanılan kalabalık benzetimleri genellikle gerek zamanda alıŐma zorunluluĐu iermemekle birlikte bu uygulamaların oĐunda kalabalıklar en ok dikkat eken birincil n plan unsurları olarak grev yaparlar. Yani benzetimin gereĐe uygunluk aısından kalitesi maliyetinden ok daha nemlidir. Bu gibi bilimsel kullanımlarda, benzetimin gereksinimlerine gre kalabalıĐın grsel ve davranıŐsal kalitesinin, benzetimin baŐında uygun biimde ayarlanarak grsel veya davranıŐsal anlamda gereĐe olabildiĐince yakın benzetimler sunulması mmkündür. DiĐer taraftan, gerek zaman kısıtı olsun veya olmasın, artistik ve estetik kalabalık uygulamalarının oĐunda kullanıcının ilgisi genellikle benzetimi yapılan kalabalık yerine daha nemli n plan karakterleri, varlıkları veya olayları zerinde yoĐunlaŐmaktadır. Bu sayede mmkn olan her yerde benzetimin kalitesi yerine performansı n planda tutularak zellikle maliyetin de nem arz ettiĐi kullanımlarda inandırıcı kalabalık benzetimleri oluŐturulması olanaklı hale gelmektedir. Bu amala kalabalıĐın genel inandırıcılıĐını korurken grselleŐtirme [1], animasyon [2] veya arpıŐma tespiti ve engellemesi [3], [4] gibi alanlarda yerel veya bunların hepsinin birden [5] genel kalitesini dŐurerek daha yoĐun ve/veya daha byk kalabalıklar oluŐturmayı amalayan detay seviyesi (DS) teknikleri geliŐtirilmiŐtir.

Bununla birlikte zellikle kalabalıĐın kullanıcı tarafından kontrol edilmediĐi veya kullanıcı girdilerine tepki vermediĐi gerek zamanlı eĐlence ve dekoratif amalı uygulamaların oĐunda kalabalık benzetimleri var olan sahne ierisinde bir arka plan oluŐturmak iin kullanılmaktadır. Bunun baŐlıca nedeni sahneye arka plan olarak sadece bir kalabalık eklenmesi durumunda bile kullanıcı tarafından algılanan gerekliĐin nemli lde artmasıdır. Kullanıcı etkileŐimi iermeksizin sadece arka plan atmosferini desteklemek iin kullanılan bu kalabalık benzetimlerinde gereki fakat maliyetli gezinim hareketlerinin gerekleŐtirilmesine de gerek yoktur. Bu gibi ambiyans kalabalıkları iin yukarıda sz edilen performans karŐılıĐında kısmi kaliteden feragat eden DS tekniklerinden faydalanılması mmkn olmakla birlikte kullanıcı ilgisi kalabalık zerinde nadiren odaklanacaĐından gezinim hareketlerinin kalitesini benzetim genelinde dŐurecek daha dŐuk maliyetli benzetim yntemlerinin tercih edilmesi daha mantıklı olacaktır. Bu amala geliŐtirilmiŐ olan ve benzetim ncesinde hesaplanmıŐ arpıŐmasız yrngelerden oluŐan yapı

taşlarını birleştirerek farklı büyüklükte benzetim ortamları ile bunlar üzerinde hareket eden kalabalıklar oluşturan “kalabalık parçaları” (*crowd patches*) yaklaşımının değişik türevleri [6]–[8] bulunmakla birlikte bunların hepsi hem parçaların hesaplanması için ön işlem maliyeti hem de hesaplanan parçaların saklanması için bellek maliyeti gerektirmektedir (Şekil 1.2).



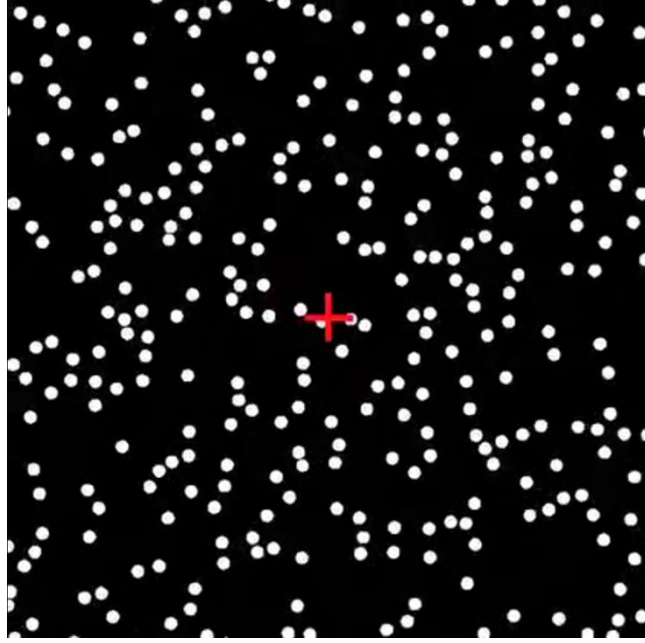
Şekil 1.2 Kalabalık parçaları yöntemiyle oluşturulmuş bir kalabalık benzetiminin örnek sahne görüntüsü [6]

Alternatif olarak, etkileşimsiz ambiyans kalabalıklarının benzetimini gerçek zamanda son derece düşük bir gezinim maliyetiyle gerçekleştirmek üzere, kalabalık sahnelerinin önceden kaydedilmiş gezinim betiklerinden faydalanılması da mümkündür. Bu betikler bireylerin hız vektörlerini uygun biçimde ayarlayarak üzerinde gezindiği çarpışmasız yollardan oluşmaktadır. Söz konusu bu çarpışmasız yolları betikler içerisinde saklamanın en kolay yöntemi ise bu yolları, üzerlerindeki hareket boyunca tek bir sabit hız vektörü kullanmanın yeterli olacağı küçük alt parçalara bölmektir. Bu sayede elde edilen yol parçalarının her biri için yalnızca başlangıç pozisyonunun, sabit hız vektörünün ve seyahat süresinin (ya da bitiş pozisyonunun) kaydedilmesi yeterli olacaktır. Gezinim betiklerinin bir insan tarafından kurgulanarak elle oluşturulması mümkündür. Ancak bu durum çok zaman alıcı ve hata yapmaya açık bir süreç olmanın yanı sıra maksimum birey sayısı için de muhtemelen düşük bir üst sınıra sahip olacaktır. Diğer taraftan, çevrimdışı benzetimler gerçekleştirerek her bir bireyin hız vektörü değişikliklerini kaydetmek suretiyle betik oluşturma sürecini otomasyon haline getirmek üzere alanda mevcut olan gezinim yöntemlerinin birçoğundan faydalanılabilir. Böylelikle önceden kaydedilmiş gezinim betikleri kullanılarak ambiyans kalabalıklarının çok düşük bir gezinim maliyeti ile gerçek zamanda yeniden üretilmesi olanaklı hale gelmektedir. Ancak bu gezinim yöntemlerinin birçoğu bireylerin sürekli

salınım gerçekleştiren hız vektörü değişikliklerinden kaynaklanan tutarsız hareketlerine neden olacağından, gerçekleşecek tüm hız vektörü değişikliklerini kaydetmek durumunda olan gezinim betiklerinin bellek gereksinimleri de gereksiz bir biçimde çok yüksek olacaktır. Dahası, önceden hesaplanan ve farklı şekillerde birleştirilerek tekrar tekrar kullanılabilen kalabalık parçalarının aksine bu betikler, belirli bir benzetim alanı üzerinde kurulu belirli bir sahneye özel olarak oluşturulduğu için farklı özelliklere sahip sahnelerde (benzetim alanlarında) tekrar kullanım esnekliğine sahip değildir.

Yakın zamanda yapılan psikolojik bir çalışmada [9] doğrusal çarpışmasız yörüngeler üzerinde sabit hızla hareket eden dairesel nesnelere (Şekil 1.3) izleyen denekler, bu nesnelere çarpışmadan kaçınma manevrası yapıyor muydu gibi görünmelerini rapor etmişlerdir. Bu çalışmadan esinlenilerek doğrusal ve çarpışmasız yörüngeler üzerinde sabit hızla hareket eden bireyler kullanılarak düşük gezinim maliyetine rağmen inandırıcı bir kalabalık oluşturmanın mümkün olabileceği öngörülmüştür. Benzetimi oluşturan bireylerin hiç manevra yapmadan ilerlemesi veya çok sınırlı bir manevra kabiliyetine sahip olması ilk bakışta gerçekçiliği olumsuz etkileyecek gibi görünmektedir. Ancak Zipf [10] tarafından ortaya konulan “en az çaba prensibi” doğrultusunda canlıların doğal olarak hedeflerine ulaşmak üzere en az çaba gerektireceklerini düşündükleri yolları kullanmayı seçeceği ileri sürülmüştür. Söz konusu bu yaklaşım doğrudan insan yürüyüşü üzerinde, harcanacak metabolik enerjiyi minimize etme şeklinde de gözlemlenmiştir [11]. Ayrıca Still [12] kalabalık dinamikleri ve davranışlarının da en az çaba prensibi ile uyumlu olduğunu çok sayıda vaka ve veri üzerinde örneklemiştir. En az çaba prensibi ile ilgili bu bilgilerden yola çıkan Guy ve arkadaşları [13], bireylerin hedeflerine ulaşmak için harcayacağı enerjiyi minimize etmek üzere olabilecek en kısa yolu kullanarak mümkün olan en az sayıda ve en azami ölçüde çarpışmadan kaçınma manevrası gerçekleştirmelerini sağlayacak bir kalabalık gezinim yöntemi geliştirmişlerdir. Bununla birlikte birey tabanlı diğer bazı gezinim metotlarının aksine, geliştirdikleri bu yöntemin bireyler için daha tutarlı manevralar gerçekleştiren, daha pürüzsüz (*smooth*) yollar oluşturabildiğini göstermişlerdir. Bu çalışmaların ışığında, kalabalık benzetimlerinde hedeflerine ulaşmak üzere doğrudan çarpışmasız yollar üzerinde ilerleyen bireylerin, inandırıcılığı olumsuz etkilemek yerine gayet doğal karşılanabileceğini söylemek çok da yanlış olmayacaktır. Tüm bu çalışma ve öngörülerden hareketle tez çalışması kapsamında, 2 boyutlu benzetim alanları üzerinde doğrusal ve çarpışmasız yörüngeler kullanarak sabit hızla hareket eden bireylerden oluşan, maksimum birey yoğunluğuna sahip etkileşimsiz ambiyans kalabalıklarının gerçek zamanda

oluşturularak düşük gezinim maliyetiyle benzetiminin gerçekleştirilmesi için gerekli çalışmalar yapılmıştır.



Şekil 1.3 Gerçekleştirilen psikolojik bir çalışmada [9] doğrusal çarpışmasız yörüngeler üzerinde sabit hızla hareket eden dairesel nesnelere

1.2. Tez Çalışmasının Temel Katkıları

Düşük gezinim maliyetine sahip, gerçek zamanlı bir kalabalık benzetimi oluşturmak için üç yeni “yönlendirmesiz” (*steering-free*) gezinim yöntemi geliştirilmiştir. Yöntemlerin her birinde de bireylere benzetime dâhil oldukları anda başlangıç noktalarından bitiş noktalarına kadar çarpışmasız olduğu garanti edilen yollar atanmaktadır. Bireyler herhangi bir çarpışma ile karşılaşmayacaklarından emin oldukları üzere benzetim süresince sabit hızlar ile herhangi bir doğrultu değişikliği yapmadan bitiş noktalarına doğru yürümektedir. Bu sayede bireylerin benzetimde kaldıkları süre boyunca, yüksek işlem maliyetine sahip herhangi bir çarpışma tespiti veya çarpışmadan kaçınma hesaplaması yapılmasına gerek olmadığından benzetimin gezinim maliyeti mümkün olan en düşük seviyeye çekilmektedir.

Yönlendirmesiz gezinim yaklaşımının yapılabilirlik (fizibilite) araştırması olarak düşünülebilecek ilk yöntemde bireylerin başlangıç noktalarından bitiş noktalarına kadar çarpışmasız, doğrusal ve sabit hızlı tek bir yörünge parçasından oluşan yollara sahip olmaları sağlanmıştır. İhtiyaç duyulan bu yolları (yörünge parçalarını) belirlemek üzere tamamı AİB üzerinde çalışan bir birey oluşturma süreci görevlendirilmiştir. Her bir doğrusal ve sabit hızlı

yörünge parçası başlangıç pozisyonu, bitiş pozisyonu ve yürüme hızı parametreleriyle ifade edilmiştir. Yeni bir çarpışmasız yörünge parçası belirlemek üzere bu parametrelerin her biri için belirli aralıklar dâhilinde rastgele değerler seçilerek karşılık gelen tek yörünge parçalı çarpışmasız yolun (TYPÇY) benzetimde kalınacak süre boyunca var olan bireylerle çarpışma içerip içermediği kontrol edilmiştir. Tez metninin devamında bu ilk yöntemden kısaca AİB tabanlı TYPÇY yöntemi olarak bahsedilecektir.

Geliştirilen ikinci gezinim yönteminde de AİB tabanlı TYPÇY yöntemine benzer biçimde bireylerin başlangıç noktalarından bitiş noktalarına kadar TYPÇY'lere sahip olmaları sağlanmıştır. Bireylere tahsis edilen TYPÇY'leri oluşturan her bir yörünge parçası sabit bir doğrultu ve sabit bir hızdan oluşan sabit bir hız vektörüne karşılık gelmektedir. Söz konusu bu çarpışmasız sabit hız vektörlerinin benzetim içerisinde gerçek zamanda bulunmasını sağlamak üzere grafik işlem biriminin (GİB) özelleşmiş donanımsal hızlandırma özelliklerinden faydalanan sofistike bir yaklaşım geliştirilmiştir. Buna göre, 2 boyutlu benzetim alanına dik olacak biçimde hayali bir zaman boyutu eklenerek oluşturulan 3 boyutlu konum-zaman uzayı içerisinde, benzetimde yer alan bireylerin sabit hız vektörleri ile hareketlerinden dolayı kaplayacakları sınırlayıcı konum-zamansal hacimler belirlenmiştir. Var olan bu hacimler yeni oluşturulacak bireyler için engel teşkil edeceğinden bu hacimlerin her birine konum-zamansal yörünge engeli (KZYE) ismi verilmiştir. Daha sonra 3 boyutlu konum-zaman uzayı içerisinde kaplanan bu KZYE hacimlerinden arda kalan boşluklar değerlendirilerek yeni oluşturulacak bireyler için uygun boş hacimler yani sabit hız vektörleri belirlenmesi sağlanmıştır. Geliştirilen ilk yöntemde benzer biçimde tez metninin devamında bu ikinci yöntemden kısaca GİB tabanlı TYPÇY yöntemi olarak bahsedilecektir.

GİB tabanlı TYPÇY yöntemi üzerine geliştirilen üçüncü metotta ise bireylerin başlangıç noktaları ile bitiş noktaları arasında birden fazla doğrusal ve sabit hızlı yörünge parçası kullanabilmesine olanak sağlanmıştır. Her birey için bu çarpışmasız yörünge parçaları, başlangıç noktasından itibaren sırayla uç uca eklenerek bitiş noktasına kadar çok yörünge parçalı çarpışmasız yol (ÇYPÇY) oluşturulmuştur. Her bir çarpışmasız yörünge parçası farklı bir sabit hız vektörüne karşılık gelmektedir. Ardışık iki yörünge parçasının hız vektörleri arasında ayarlanabilir ön tanımlı limitler dâhilinde fark olmasına izin verilerek parçaların uç uca eklendiği birleşim noktalarında doğrultu ve/veya hız değişiklikleri yapılmasına imkân tanınmıştır. Böylelikle 3 boyutlu konum-zaman uzayı içerisinde yeni oluşturulacak bireyler için tek parça bir KZYE aramak yerine küçük alt KZYE'lerden oluşan

bir bütünlük KZYE bulma özgürlüğü kazanılarak yeni bireyler için çarpışmasız yol bulma olasılığı ile buna bağlı olarak ulaşılabilecek maksimum birey sayısı ve maksimum benzetim yoğunluğu artırılmıştır.

Ayrıca üçüncü yöntemde çarpışmasız yörünge parçalarının bitiş noktalarını hesaplarken başlangıç noktasından sabit doğrultu ve hız ile gidilebilecek en uzak noktayı hızlı ve kolay bir biçimde bulmak için GİB'nin donanımsal derinlik testi özelliğinden yararlanılmıştır. Aynı hız vektörünü kullanarak, tespit edilen bu bitiş noktasından daha ileriye gitmeye çalışmak var olan bir birey ile çarpışmaya neden olacağından, bu noktadan sonra farklı bir hız vektörüne sahip yeni bir yörünge parçası aranacaktır. Birleşim noktalarında gerçekleştirilen bu hız vektörü değişiklikleri, çarpışmadan kaçınma manevrası olarak algılanarak benzetimin inandırıcılığına da katkıda bulunacaktır. Geliştirilen diğer yöntemlere benzer olması açısından tez metninin devamında bu üçüncü yöntemden kısaca ÇYPCY yöntemi olarak bahsedilecektir.

Tez kapsamında geliştirilen üç yönlendirmesiz gezinim yöntemi de benzetimi oluşturan bireyler için çarpışma tespiti ve çarpışmadan kaçınma manevraları yapmak adına maliyetli hesaplamalar gerçekleştirmeye ihtiyaç duymadığı için ana işlem birimi (AİB) üzerindeki gezinim işlemleri yükünü çok büyük oranda azaltmaktadır. Bunu sağlayabilmek adına kullanılan çarpışmasız olduğu garanti edilmiş yolları oluşturan, doğrusal ve sabit hızlı yörünge parçalarını GİB aracılığıyla belirleyebilen yöntemler sayesinde, kalabalık benzetimlerinde çoğu zaman görselleştirme işlemleri haricinde kullanım dışı olan GİB'nin potansiyel işlem gücünden de faydalanılması sağlanmıştır. Böylelikle AİB'nin işlem gücünde tasarrufa ihtiyaç varken, GİB'nin işlem kapasitesinin görece az bir bölümünün kullanılacağı kalabalık benzetimi ihtiyaçları için faydalanılması muhtemel gezinim çözümleri sunulmuştur.

Ayrıca geliştirilen yönlendirmesiz gezinim yöntemlerin her birinde, sınırlı sayıda yörünge parçasından oluşan çarpışmasız yollar üzerinde hareket eden bireylerin, en az çaba prensibi ile uygun bir biçimde en kısa yoldan hedeflerine ulaşmak adına yalnızca gerekli olduğu anlarda ve gerekli olduğu ölçüde hız vektörü değişiklikleri gibi manevralar yapacakları garanti edilmiştir. Bu sayede son derece tutarlı, çarpışmasız hız vektörleri ile oldukça pürüzsüz yollar üzerinde daha gerçekçi bir biçimde hareket eden bireylerden oluşan, daha inandırıcı kalabalık benzetimleri gerçekleştirilmesi sağlanmıştır. Tüm bunlara ek olarak, önceden kaydedilmiş gezinim betiklerinin otomatik bir şekilde oluşturulması sürecinde, sınırlı sayıda hız vektörü değişikliği manevrası gerçekleştiren bu tip bireylerden oluşan

benzetimlerden yararlanılması sayesinde, bellek maliyeti önemli ölçüde azaltılmış gezinim betikleri oluşturulması da olanaklı hale getirilmiştir.

1.3. Tez Metninin Organizasyonu

Tez metninin devamında kullanılan kesimler ve bunların tasarımı ana hatlarıyla şu şekildedir:

İkinci bölümde öncelikle kalabalık, kalabalık benzetimi ve kalabalık benzetiminin kullanım alanları hakkında alan bilgisi verilmiştir. Daha sonra kalabalık benzetiminin bileşenleri ile ilgili bilgi verilmiş ve bu alanda yapılan çalışmaları ele alan bir alanyazın özeti sunulmuştur. Son olarak, kalabalık benzetiminde kullanılan detay seviyesi teknikleri ile kalabalık benzetimlerinin inandırıcılığı konularına değinilerek yapılan çalışmalardan bahsedilmiştir.

Üçüncü bölümde yönlendirmesiz gezinim yaklaşımının ilk adımı olan AİB tabanlı TYPÇY yönteminde her bir bireyin sahip olduğu TYPÇY'nin AİB yardımıyla belirlenme süreci hakkında bilgi verilmiştir. Sonrasında ise bireylerin bu TYPÇY'ler üzerinde düşük maliyetle geziniminin ayrıntıları anlatılmıştır.

Dördüncü bölümde ilk olarak TYPÇY bulma problemi konum-zamansal açıdan ele alınmıştır. Daha sonra bu konum-zamansal probleme getirilen GİB destekli çözüm yaklaşımı açıklanmıştır. Devamında ise tek parça çarpışmasız, doğrusal ve sabit hızlı yörüngeler kullanan kalabalık gezinim yöntemine dair ayrıntılara yer verilmiştir.

Beşinci bölümde birden fazla doğrusal ve sabit hızlı yörünge kullanarak bireyler için çarpışmasız yollar oluşturmanın detayları açıklanmıştır. Ayrıca ÇYPÇY'ler kullanan kalabalık gezinim yönteminin gerçekleştirdiği gezinim işlemleri ile ilgili bilgi verilmiştir.

Altıncı bölümde geliştirilen üç yöntemin birey oluşturma başarımını tüm ayrıntılarıyla değerlendirmek adına yapılan deneyler ve sonuçları sunulmuştur. Ek olarak, geliştirilen bu yöntemlerin alanda var olan popüler bir gezinim yöntemiyle birey oluşturma ve gezinim maliyetleri açısından karşılaştırılması için düzenlenen deneylere değinilmiş ve sonuçlar ayrıntılı bir biçimde irdelenmiştir. Dahası, geliştirilen gezinim yöntemlerinin sabit engeller içeren benzetim alanları üzerindeki birey oluşturma başarımına ilişkin deneyler ve sonuçları paylaşılmıştır. Son olarak da bu yöntemlerin yörünge kalitesinin alanda var olan popüler bir gezinim yöntemiyle kıyaslanarak değerlendirilmesi yapılmıştır.

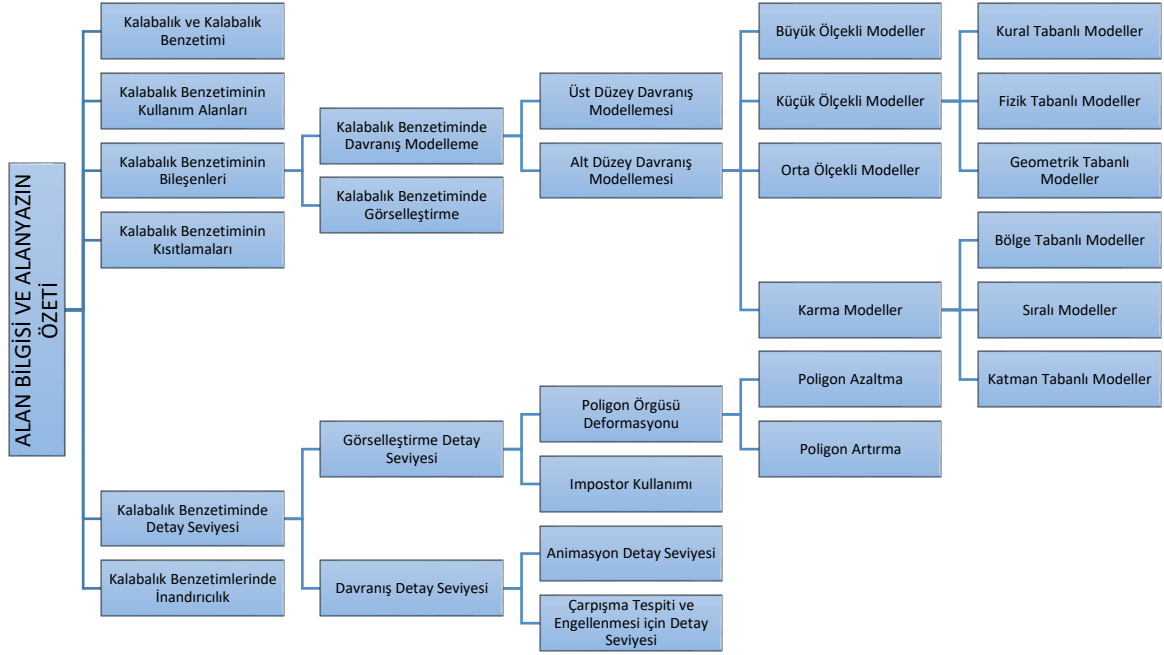
Yedinci bölümde geliştirilen yöntemlerin sonuçları değerlendirilerek olumlu ve olumsuz yönleri tartışılmış, bu yöntemlerin potansiyel kullanıcılarına yol gösterebilecek nitelikte

birtakım kullanım önerilerinde bulunulmuş ve gelecekte yapılabilecek çalışmalara yön verecek, olası araştırma problemlerinden bahsedilmiştir.

Sekizinci ve son bölümde ise tezde yapılan çalışmalar ile bunların genel sonuçları ve katkıları kısa ve öz bir biçimde ifade edilmiştir.

2. ALAN BİLGİSİ VE ALANYAZIN ÖZETİ

Bu bölümde, Şekil 2.1’de gösterilen hiyerarşik yapıya uygun olarak kalabalık, kalabalık benzetimi, kalabalık benzetiminin kullanım alanları, kalabalık benzetiminin bileşenleri, kalabalık benzetiminde kullanılan detay seviyesi teknikleri ile kalabalık benzetimlerinin inandırıcılığı gibi konulara değinilerek alan bilgisi verilmiş ve alanda bu konularla ilgili yapılmış çalışmaları ele alan bir alanyazın özeti sunulmuştur.



Şekil 2.1 Alan bilgisi ve alanyazın özetinde ele alınan başlıkların hiyerarşik yapısı

2.1. Kalabalık ve Kalabalık Benzetimi

Kalabalık, bireylerin ortak amaçları paylaşımları nedeniyle aynı fiziksel ortamda toplanarak oluşturduğu bir grup olarak tanımlanmaktadır [14]. Günlük yaşantımızda genellikle insan topluluklarından oluşan kalabalıklarla karşılaşmamıza rağmen, hayvan sürüleri, böcek kümeleri, araç trafikleri hatta ve hatta robot gruplarının oluşturduğu kalabalıklardan söz etmek mümkündür [15]. Hangi tür bireylerden oluşursa oluşsun, kalabalıklar basitçe bireylerden oluşan topluluklar olmaktan ziyade kendi içlerinde karmaşık dinamikleri olan sistemlerdir. Bireylerin bir araya gelerek oluşturdukları kalabalıklar içerisindeki kitlesel davranışları, yalnızken gerçekleştirdikleri davranışlardan çok farklıdır. Kalabalık içerisindeki bireylerin davranışları çeşitli fizyolojik, psikolojik ve sosyal faktörlere bağlı olarak kalabalığı oluşturan diğer bireylerin davranışlarından etkilenmektedir [16]. Duruma bağlı olarak bireyler kalabalık için uygun görünen davranışları istemsiz olarak sergilemek

zorunda kalabilir. Örneğin; tribünde maç izleyen bir seyirci etrafındaki diğer bireylerin tezahürata başlamasıyla takımını desteklemek adına kendisini bu tezahürata katılmak zorunda hissedebilir.



Şekil 2.2 Gerçek bir kalabalık fotoğrafı ve bu kalabalığın benzetimi yapılan sanal gerçekleştirimi [17]

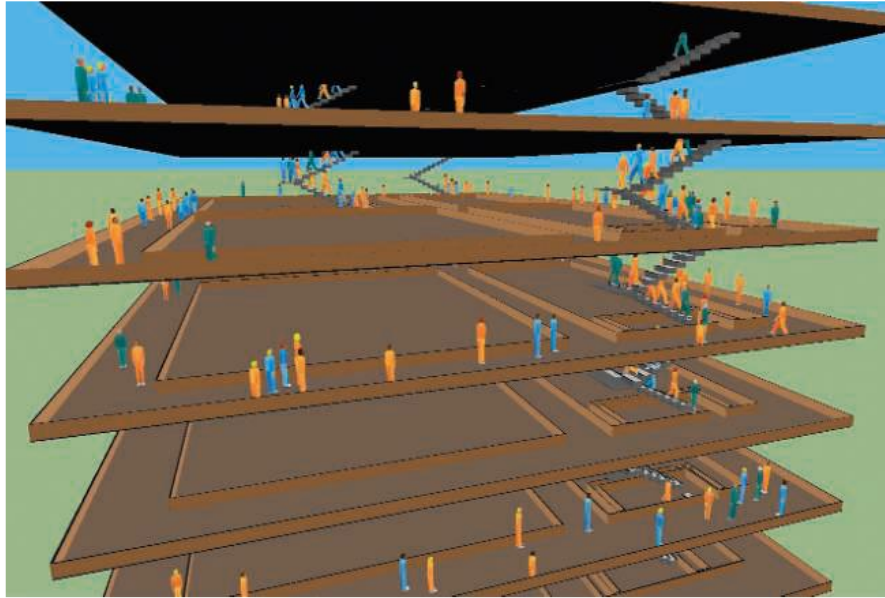
Kalabalık benzetiminin birçok tanımı bulunmakla birlikte genel olarak kalabalık benzetimi, çok sayıda bireyin oluşturduğu toplulukların hareket ve davranışlarının gerçeğe uygun olarak Şekil 2.2’de örneklendiği gibi taklit edilmesidir [18]. Kalabalık benzetiminin, kişisel zekâya sahip çok sayıda bireyin sanal bir sahne içerisine yerleştirilerek farklı kitlesel davranışlar sergilemesi şeklinde tanımlanması da mümkündür [19]. Burada belirtildiği üzere kalabalığı oluşturan bireylerin kendilerine has düşünme ve karar verme yeteneğinin bulunması, bireylerin hareket ve davranışlarına olağanüstü bir çeşitlilik getireceği için kalabalık benzetiminin gerçeğe uygun olması açısından büyük önem taşır. Ancak kalabalığı oluşturan her bir birey için diğerlerinden tamamen farklı bir zekâ modellemesi yapılması her zaman mümkün olmayabilir. Özellikle gerçek zamanlı benzetimlerde birey sayısının artmasıyla birlikte gelen yüksek işlem gücü ihtiyacı, koşturucu işlem kabiliyeti olan donanım ve/veya algoritmalar için dahi sınırlayıcı bir unsur haline gelmektedir. Bu nedenle kalabalık benzetimlerinde, bireye özgü zekâ modellemesi yapmak yerine bireylerin tümü için ortak olarak kullanılan, ancak bireylerin farklı içsel ve/veya dışsal(çevresel) etkenlere göre farklı durumlarda farklı kararlar verebilmesine olanak veren modellemeler kullanılmaktadır.

2.2. Kalabalık Benzetiminin Kullanım Alanları

Günlük hayatımızda değişik türde bireylerin oluşturduğu çok farklı kalabalıklar ile karşılaşırız. Bu kalabalıkların nasıl oluştuğunu, hangi hareket ve davranışlarda bulunduğunu inceleyen çok çeşitli araştırma alanları mevcuttur. Ancak bu araştırmalar için insan, hayvan,

araç vb. gerçek bireylerden kalabalıklar oluşturmak ve kullanmak her zaman mümkün olmayabilir.

Öncelikle gerçek bireylerden oluşan bir kalabalık kullanmak tehlikeli olabilir. Örneğin; bir savaş taktiği kurgusunun gerçek insanlar ile hayata geçirilmesi insan hayatı açısından çok büyük risk taşır. Benzer şekilde, güvenlik biliminde doğal afet, yangın veya terör kaynaklı olası bir acil durumda büyük yapılarda bulunan insanların güvenli tahliyesi için gerçek insan kalabalıklarıyla çalışmak insan hayatı, insan sağlığı, mal mülk güvenliği ve hatta çevre açısından riskli durumlar oluşturabilir. Şekil 2.3'te bir örneği verilen bu gibi acil durum senaryolarında kalabalık benzetimi hiçbir risk taşımadığından sıklıkla kullanılmaktadır [20]–[24].



Şekil 2.3 Acil durum boşaltma senaryolarında kalabalık benzetimi kullanımı [24]

Bir diğer etken olarak, gerçek bireylerden oluşan kalabalıklar üzerinde araştırma yapmak, zamansal ve parasal açıdan çok yüksek maliyetleri de beraberinde getirebilir. Örneğin; yoğun trafiğe sahip bir bölgenin trafiğini düzenlemek adına gerçek araçlarla kalabalık oluşturulması durumunda, senaryonun her bir tekrarı için çok fazla zaman ve para ihtiyacı olacaktır. Bu nedenle mimari tasarım [25]–[27], psikoloji [28], robotik [29], sosyoloji [30], [31], şehir planlama [32], trafik planlama [33]–[38] (Şekil 2.4) gibi alanlarda kalabalık benzetimi etkin bir şekilde kullanılmaktadır.



Şekil 2.4 Trafik sinyal ve işaretleri bulunan dört yönlü bir kavşaktaki araç trafiğinin benzetimi [35]

Bazı durumlarda ise gerçek bireylerden oluşan kalabalığı yapılacak araştırma doğrultusunda yönlendirmek mümkün olmayabilir. Örneğin; hayvan sürüleriyle yapılacak araştırmalarda hayvanların hareket ve davranışları üzerinde mutlak kontrole sahip olmak mümkün olmayacaktır. Bu sebeple fizik [39], hayvan sürüsü benzetimleri [40]–[43] (Şekil 2.5) vb. alanlarda da kalabalık benzetimi kullanılmaktadır.



Şekil 2.5 Büyük bir göçmen çekirge sürüsünün küçük bir köyü istilasının benzetimi [42]

Bunların dışında doğası gereği sanal sahneler içeren bilgisayar oyunları, sanal gerçeklik uygulamaları ve görsel efektler içeren filmler gibi eğlence sektöründe de kalabalık benzetimleri sıklıkla kullanılmaktadır. Eğlence alanında kalabalık benzetiminin kullanımını kolaylaştırmak adına çeşitli ticari uygulamalar geliştirilmiş ve bu uygulamalar birçok yapımda başarıyla kullanılmıştır. Örneğin; Massive yazılımı [44] The Lord of the Rings, Avatar gibi birçok filmde kullanılmıştır. Yine bu uygulamalardan biri olan Golaem Crowd [45], Game of Thrones, Hercules gibi yapımlarda kullanılmıştır.

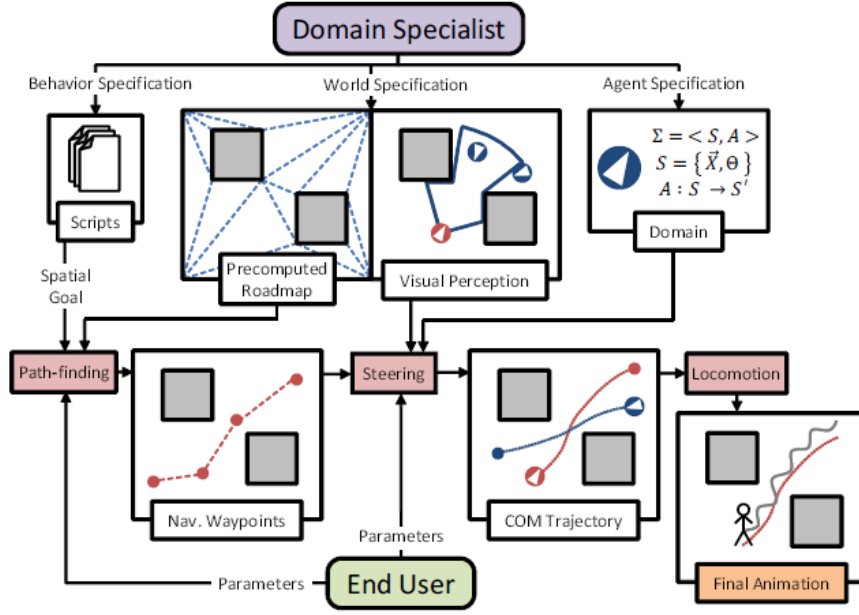
2.3. Kalabalık Benzetiminin Bileşenleri

Kalabalık benzetimi temelde iki bileşen altında incelenebilir. Bunlardan ilki benzetimi yapılan kalabalığın davranış modellemesidir. Bu bileşen kalabalığı oluşturan bireylerin gerçek dünyadaki hareketlerini taklit ederek bireylere hareket etme kabiliyeti kazandırılmasını sağlar. İkincisi ise kalabalığın görselleştirilmesidir. Görselleştirme kapsamında benzetimi yapılan kalabalığı oluşturan bireylerin gerçeğe olabildiğince yakın görüntülenmesi sağlanmaya çalışılır.

Sözü edilen bu iki bileşenden oluşan kalabalık benzetimleri, her zaman en gerçekçi davranış modellemesine ve en kaliteli görselleştirmeye ihtiyaç duymayabilir [15], [46]. Acil durum tahliye senaryoları, trafik planlama vb. ihtiyaçlar için kullanılan kalabalık benzetimlerinde, benzetimin görsel kalitesinden çok gerçeğe uygunluğu ön planda olduğu için bu tür benzetimlerde 2 boyutlu basit bir görselleştirme, kullanıcıların benzetimi daha iyi takip edilebilmesi için yeterli olmaktadır. Tam aksine film yapımları ve bilgisayar oyunlarında ise izleyicilerin ve oyuncuların odağı çoğunlukla benzetimin görsellerinde olacağından bu tür benzetimlerde inandırıcı görsel efektlere ulaşmak adına yüksek kaliteli görselleştirme teknikleri kullanılmaktadır.

2.3.1. Kalabalık Benzetiminde Davranış Modelleme

Davranış modelleme, kalabalığı oluşturan bireylerin gerçekteki hareket ve davranışlarının benzetim içerisine uyarlanarak bireylere gerekli hareket ve davranışları sergileme yeteneğinin sağlanmasıdır. Geçmişten günümüze kalabalık benzetimlerinde daha ikna edici davranışlar üretmek adına ciddi miktarda çaba sarf edilmiş olmakla birlikte alanda bu konu ile ilgili detaylı alanyazın taramaları [15], [16], [47] bulunmaktadır. Kalabalık benzetimi bağlamında yapılan ve Şekil 2.6'da gösterildiği üzere çok farklı öğelerden oluşan davranış modellemesini katmanlı bir yapıda düşünerek üst düzey ve alt düzey davranış modellemesi olmak üzere iki ana başlık altında incelemek mümkündür.



Şekil 2.6 Bireylerin otonom varlıklar olarak modellenmesi için kullanılan geleneksel yaklaşım [48]

2.3.1.1. Üst Düzey Davranış Modellemesi

Üst düzey davranışlar arasında benzetim ortamı içerisinde boş boş dolaşma, bir nesneden veya bir bölgeden uzaklaşma, bir nesneyi takip etme, kuyruk oluşturma, grup halinde hareket ederken belirli bir formasyon oluşturma gibi örnekler sayılabilir. Üst düzey davranış modelleme, benzetim içerisinde yer alan bireylerin “ne” yapmaları gerektiğine karar veren katmandır. Karar verme işleminde bireylerin konum, hız, doğrultu gibi fiziksel özellikleri ile duygu durum gibi duygusal özelliklerine ek olarak geniş bir çevrelerinde yer alan diğer faktörler de etkili olabilmektedir.

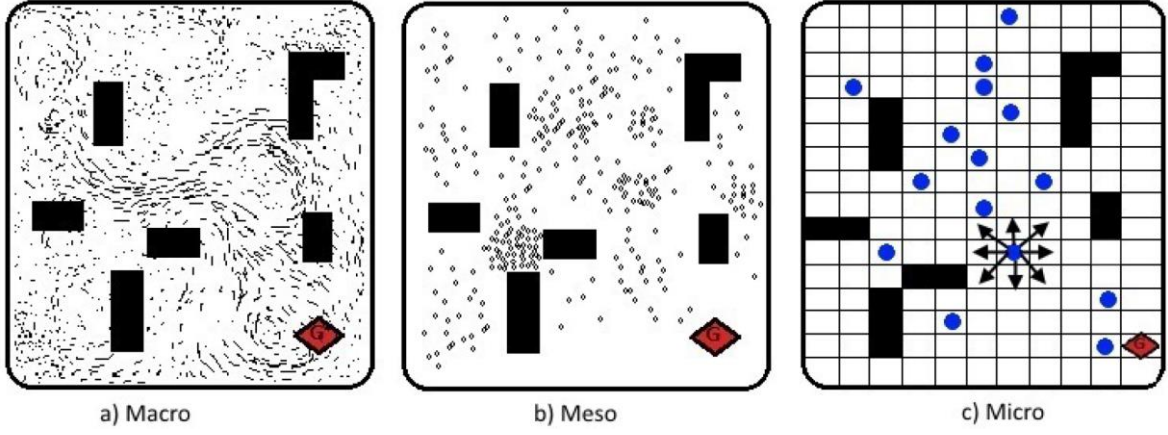
Üst düzey davranışları oluşturmanın ilk yolu ön tanımlı betikler kullanmaktır [49]. Bu betikler sayesinde çok karmaşık davranışların çok hassas bir şekilde benzetim içerisinde gerçekleştirilmesi sağlanabilir. Farklı davranışları gerçekleştirebilmek için farklı betiklerin tanımlanmasını gerektiren bu yöntemde, oluşturulan bu betikler benzetim ortamına özgü olacağı için başka ortamlarda kullanılamayabilir. Bu durum her ortam için ihtiyaç duyulan betiklerin uyarlanması veya yeniden yazılmasını gerektirebilir. Ayrıca ön tanımlı bu betiklere göre hareket eden bireyler, yeterince otonom davranıyor hissi oluşturamayacağı gibi benzer koşullarda sürekli aynı davranışların sergilenmesine neden olduğundan benzetimi oluşturan bireyleri çok daha tahmin edilebilir bir hale getirecektir.

Bu davranışları oluşturmanın bir diğer yolu ise bireylerin çevrelerinden aldıkları geri bildirim doğrultusunda otonom bir şekilde davranmasını sağlamaktır [49]. Bu sayede daha uyarlanabilir, daha otonom ve daha az tahmin edilebilir bir benzetim oluşturmak mümkün olsa da beraberinde gelecek işlem maliyetini dengelemek adına bireylerin sergilediği davranışların kalitesinde bir azalma söz konusudur. Sadece bu tür otonom üst düzey davranışları oluşturmak için özel olarak geliştirilmiş bir yöntem bulunmamakla birlikte bu davranışlar çeşitli alt düzey davranış modellemelerinin istemli veya istemsiz sonuçları olarak elde edilebilmektedir.

2.3.1.2. Alt Düzey Davranış Modellemesi

Alt düzey davranışlara örnek olarak kalabalık davranışlarının temel bir ögesi olan ve çok sayıda araştırmacının ilgisini çeken [48], [50], [51] gezinim, bir diğer deyişle bulunulan konumdan hedef noktasına doğru hareket etme ve bunu yaparken olası çarpışmaları tespit ederek bu çarpışmalardan kaçınma işlemleri verilebilir. Alt düzey davranış modelleme, bireylerin benzetim içerisinde aldıkları üst düzey kararları yerel olarak “nasıl” uygulayacaklarını belirler. Alt düzey davranış modelleme sürecinde bireylerin kişisel özellikleri ile yakın çevrelerinin özellikleri dikkate alınarak kısa süre için geçerli olacak yerel kararlar verilir. Bu nedenle benzetim boyunca alt düzey davranış modelleme süreci genellikle üst düzey davranış modelleme sürecine göre daha sık aralıklarla tekrarlanır.

Kalabalık benzetiminde gezinim üzerine çok sayıda çalışma yapılmış olup Şekil 2.7’de gösterildiği gibi kimi kaynaklarda bu çalışmaların büyük ölçekli (makroskobik), orta ölçekli (mezoskopik) ve küçük ölçekli (mikroskobik) olmak üzere üç ana başlık altına gruplandırıldığı görülmüştür [51], [52]. Ancak çoğu kaynakta orta ölçekli gezinim yöntemlerinin küçük ölçekli sınıfına dâhil edilmesiyle büyük ölçekli ve küçük ölçekli olmak üzere iki grupta bir sınıflandırma yapıldığını söylemek mümkündür [14]–[16], [46], [53]. Ayrıca büyük ve küçük ölçekli modelleri bir arada kullanan karma modellerden de bahsedilebilir. Söz konusu gezinim yöntemi grupları arasındaki farkın daha detaylı anlatılabilmesi adına tez metninin ilerleyen bölümünde anlatıma büyük, orta ve küçük ölçekli modellerin yanı sıra karma modeller de dâhil edilerek dört grupta bir sınıflandırma yapılarak devam edilecektir.



Şekil 2.7 Kalabalık benzetimi yöntemlerinin sırasıyla büyük ölçekli (makroskobik), orta ölçekli (mezoskopik) ve küçük ölçekli (mikroskobik) olmak üzere üç ana başlık altına gruplanması [51]

2.3.1.2.1. Büyük Ölçekli Modeller

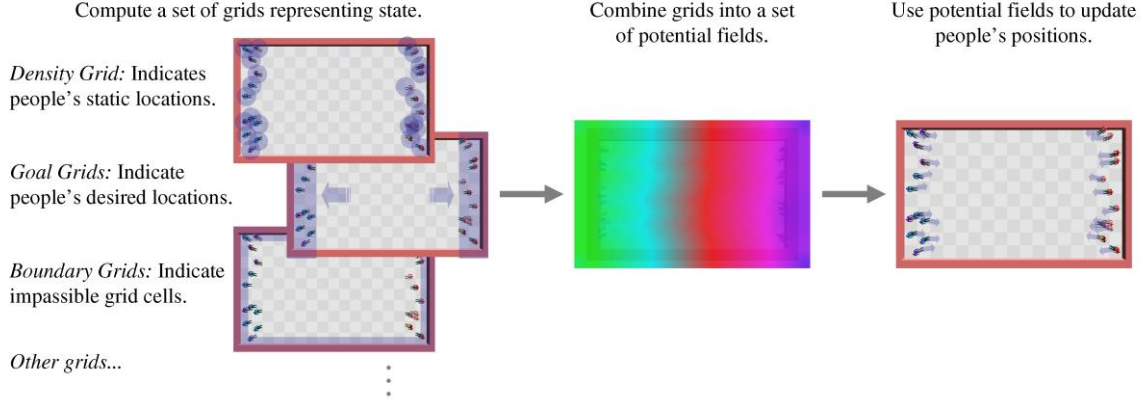
Büyük ölçekli modellerde, kalabalığı oluşturan bireylerin tamamı bir bütün olarak ele alınır. Bu modellerde her bir birey için tek tek bir sonraki hareketin (hız değişikliği, açı değişikliği) ne olacağına ve miktarına dair karar vermek yerine bireylerin, kalabalığın geneli için alınan kararlara uygun hareket etmesi sağlanarak bir bütünlük oluşturulur. Bu sayede önemli ölçüde kaynak tasarrufu sağlanması mümkün olmakla birlikte binlerce, on binlerce hatta yüz binlerce bireyden oluşan kalabalıkların benzetiminin yapılması da olanaklı hale gelmektedir. Buna karşın, büyük ölçekli modellerde alınan kararlar, bireylere özgü olmaktan çok kalabalığın geneline uygun olduğu için bireylerin, kendi doğasına özgü ince hareket ve davranışları, kişisel karakteristik özelliklerini ve çevresindeki diğer bireylerle etkileşimleri sergilemede yetersiz olduğunu söylemek mümkündür. Bu durum, özellikle yakın plan görüntülendiğinde, benzetimin gerçekçiliğini önemli ölçüde azaltmaktadır. Bunun temel nedeni büyük ölçekli modellerde bireylerin hareket edebilir alt parçaları, duygu ve düşünceleri olmayan atomik parçacıklar olarak ele alınmasıdır. Her bir birey adeta bir atom veya molekül tanesi ya da bir parçacık gibi kabul edilerek bilinen veya uyarlanan akışkanlar fiziği kurallarına göre hareket etmeleri sağlanır.

Helbing, gaz moleküllerinin kinetikleri için geliştirilmiş modelleri, yürüyen insanlara uyarlayarak kitlesel hareketlerini akışkan dinamiği şeklinde tanımlamıştır [54]. Bu sayede kalabalığı oluşturan bireylerin; farklı yönlere doğru hareket içeren alanlardan kaçınmalarını, karşıt yönlerde doğru hareketlerde her hareket yönüne özel şerit oluşturmalarını, bu durumun

tipik örneđi olan dar geçitlerden sırayla geçmelerini, birbirlerine göre hızlarını ayarlayarak çeşitli formasyonları oluşturmalarını, engellerden ve çok yüksek yoğunluklu bölgelerden kaçınmalarını sağlayarak şehir ve trafik planlama gibi uygulamalarda kullanılabilir bir kalabalık benzetimi modeli oluşturmuştur.

Hughes, kalabalığın genel davranışlarını kalabalığın yoğunluđuna bađlı sürekli bir potansiyel alanı kullanarak modellemiştir [55], [56]. Yüksek yoğunluklu ve düşük yoğunluklu olmak üzere iki farklı akış rejimi tanımlayan Hughes, bu sayede yoğunluđa bađlı potansiyel alanı kullanarak kalabalığı oluşturan bireyleri hedeflerine dođru en uygun şekilde götürmeye çalışmıştır.

Treuille ve arkadaşları, bireylerin hareketini parçacıkların enerjilerinin en aza indirgenmesi olarak ele almışlardır [57]. Hughes tarafından ortaya konan bu modeli temel alan Treuille ve arkadaşları, dinamik potansiyel alanını oluştururken yalnızca birey yoğunluđunu ve sabit engelleri deđil, kalabalığı oluşturan bireylerin hızlarını da hesaba katarak (Şekil 2.8) kalabalık içerisindeki hareket akışlarının da bireylerin hareketlerini belirlemede etkili olmasını sağlamışlardır. Bu amaçla potansiyel alanı üzerindeki eğim vektörlerini kullanarak hedefe giden en uygun yolu bulmanın yanı sıra yerel çarpışma engelleme işlemlerini de aynı anda yapabilmışlerdir. Bu sayede hem kalabalığın akışının istikrarlı olmasını hem de karşılıklı yönlerde ilerleyen bireylerin şerit formasyonu, dört yönden gelip karşı yöne gitmeye çalışan grupların girdap oluşturması gibi gerçek hayatta gözlemlenen davranışların kalabalık benzetimine dâhil edilmesini sağlamışlardır. Treuille ve arkadaşlarının geliştirdiđi bu yöntemde bireylerin yerel minimumda sıkışması engellenmiş, Hughes tarafından geliştirilen yöntemde göre bireylerin hareketlerindeki salınımlar azaltılmıştır. Buna rağmen, söz konusu yöntem ortak hedefe dođru hareket eden gruplar halindeki bireylerin benzetimi için uygun iken her bireyin farklı hedefe gittiđi senaryolarda çok yüksek işlem maliyetlerinden kaynaklı performans sorunları yaşamaktadır.



Şekil 2.8 Treuille ve arkadaşları [57] tarafından geliştirilen büyük ölçekli modelin genel algoritması

Sewall ve arkadaşları [33], trafikteki araçların benzetimi için akış tabanlı bir model kullanmışlardır. Bu modelde yollar üzerindeki her şeritte araçların tek yöne doğru hareketine izin verildiği akış tabanlı bir benzetim yapılmaktadır. Ayrıca araçların şerit değiştirmeleri gibi günlük hayatta karşılaşılan olağan bir durumu da ele almışlardır.

2.3.1.2.2. Küçük Ölçekli Modeller

Küçük ölçekli modellerde, benzetimi oluşturan bireylerin her biri ayrı ayrı ele alınarak bir sonraki hareket ve davranış hakkında bireye özgü kararlar alınması sağlanır. Her birey için tek tek işlem yapılması sebebiyle büyük ölçekli modeller gibi çok sayıda bireyden oluşan benzetimleri gerçekleştirmek mümkün olmasa da bireylerin hareket ve davranışlarındaki çeşitlilik büyük ölçekli modellere göre oldukça fazladır. Öyle ki, her bireyin farklı bir karaktere sahip olmasını ve kendi fiziksel ve duygusal özelliklerine ek olarak yakın çevresinde değişiklik gösteren fiziksel, sosyal ve psikolojik faktörlere göre hareket ve davranışlarını düzenlemesini sağlamak olasıdır. Bu durum benzetimin yakın plan görüntülediği durumlarda gerçekçilik algısını çok yüksek seviyelere çıkarabilmektedir. Küçük ölçekli modelleri çok farklı alt gruplara bölmek mümkün olmakla birlikte bu grupların başlıcaları ve temel örnekleri aşağıda ayrıntılarıyla anlatılmıştır.

2.3.1.2.2.1. Kural Tabanlı Modeller

Kural tabanlı modellerde, bireylerin hareketlerini tayin etmek adına çeşitli kurallar oluşturulur. Bu kurallar basit “eğer ... yoksa ...” koşullu ifadeleri biçiminde olmasına rağmen olası karar parametrelerinin çok farklı kombinasyonlarını kontrol ederek çok çeşitli davranışların gerçekleştirilmesine olanak tanıyabilir. Bu durum ihtiyaç duyulan kimi özel

davranışlara özgü kuralların yazılmasını gerektirebileceği için kural sayısı ve çeşitliliğinin davranış çeşitliliğini etkilediği söylenebilir. İstenilen davranışa özgü kuralları yazmak çok karmaşık olabileceği için alan uzmanlarının yardımına ihtiyaç duyulabilir. Yine de kural tabanlı yöntemler ile arzu edilen davranışa en yakın sonuçların elde edilmesi mümkündür.

Reynolds tarafından hayvan sürülerinin hareketlerini modellemek için geliştirilen kural tabanlı yaklaşım küçük ölçekli modellerin ilk önemli örneklerindedir [58]. Bu çalışmada Şekil 2.9'da örneklenen sürüleri oluşturan hayvanların hareketleri üç temel kural ile sağlanır. Bunlardan ilki bir birey ile sürüyü oluşturan diğer hayvanlar arasındaki mesafeyi koruyarak olası çarpışmaları engellemeyi amaçlar. İkincisi ise tam tersine bireyi kalabalığa yakın tutarak sürünün dağılmaması için çabalar. Sonuncu kural ise kalabalığı oluşturan bireylerin hareket doğrultularını birbirlerine göre ayarlayarak sürünün uyum içerisinde aynı yöne doğru hareket etmesini sağlar. Daha sonra Reynolds olası davranış kümesini genişleterek serbest gezinme, yol takip etme, lider takip etme gibi birçok üst düzey davranışın gerçekleştirilebilmesine olanak sağlamıştır [59].

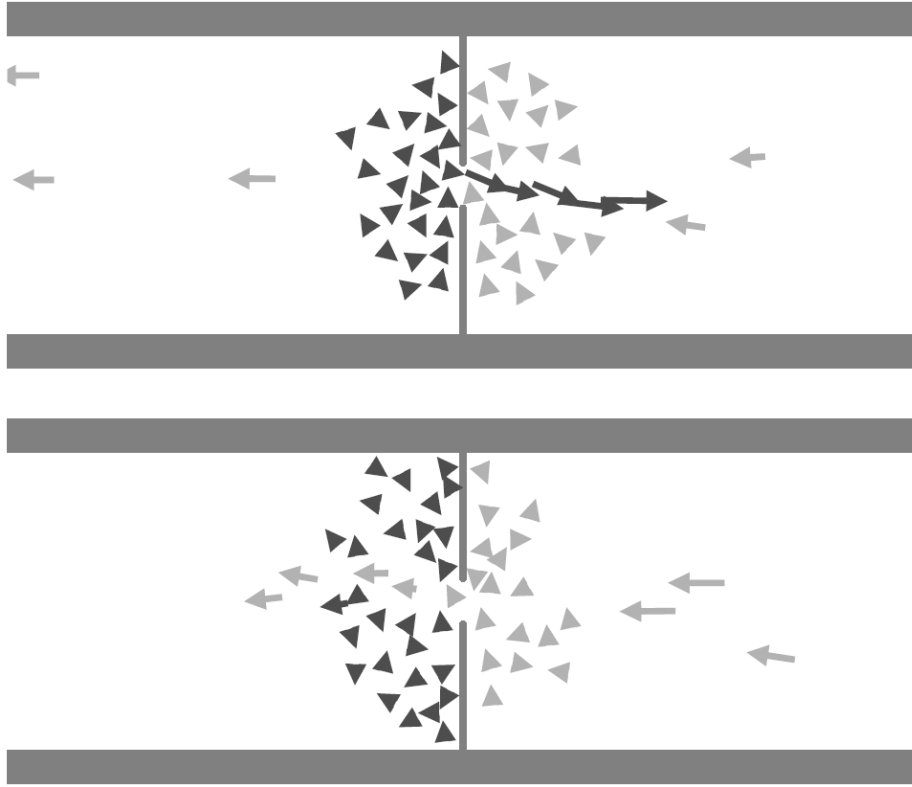


Şekil 2.9 Reynolds [58] tarafından geliştirilen kural tabanlı yöntem ile oluşturulmuş hayvan sürüsü benzetimi

2.3.1.2.2.2. Fizik Tabanlı Modeller

Fizik tabanlı modellerde bireyler kendilerine etki eden kuvvetlere göre hareket ederler. Bireylere etki eden kuvvetler, fiziksel kuvvetler olabileceği gibi sosyal, kültürel ve psikolojik kuvvetler de olabilir.

Helbing ve Molnar, bireylere etki eden sosyal kuvvetler tanımlayarak bireylerin bu kuvvetlerin bileşkesine göre ivmelenmesini sağlamıştır [60]. Bireylere etki eden kuvvetler çekici ve itici kuvvetler olarak ikiye ayrılır. Bireyin hedef noktası bireye çekici bir kuvvet (bireyin konumundan hedef noktasına doğru) uygularken benzetimde yer alan diğer bireyler ve sabit engeller bireye itici kuvvetler (diğer bireyden veya sabit engelden bireye doğru) uygular. Sonuçta elde edilen ivme vektörüne göre hız vektörü değiştirilerek kimi durumlarda yavaşlamaya kimi durumlarda ise hızlanmaya neden olmaktadır. Ayrıca Helbing ve Molnar, fizik tabanlı bu yöntemin, aynı yönde hareket eden bireylerin şerit formasyonu oluşturması, Şekil 2.10'da görülebildiği üzere iki farklı yönde hareket eden bireylerin dar bir geçitte yer alan kapıyı sırayla kullanması gibi planlanmamış, kendiliğinden ortaya çıkan davranışları sağlayabildiğini gözlemlemişlerdir [61]. Helbing ve arkadaşları [23], sosyo-psikolojik ve fiziksel kuvvetlerin birleşimini kullanarak panik durumlarında bireylerin kaçış senaryolarının benzetimini gerçekleştirmişlerdir.



Şekil 2.10 Dar bir geçitte yer alan kapıyı sırayla iki farklı yöne hareket eden bireylerin kullanması [61]

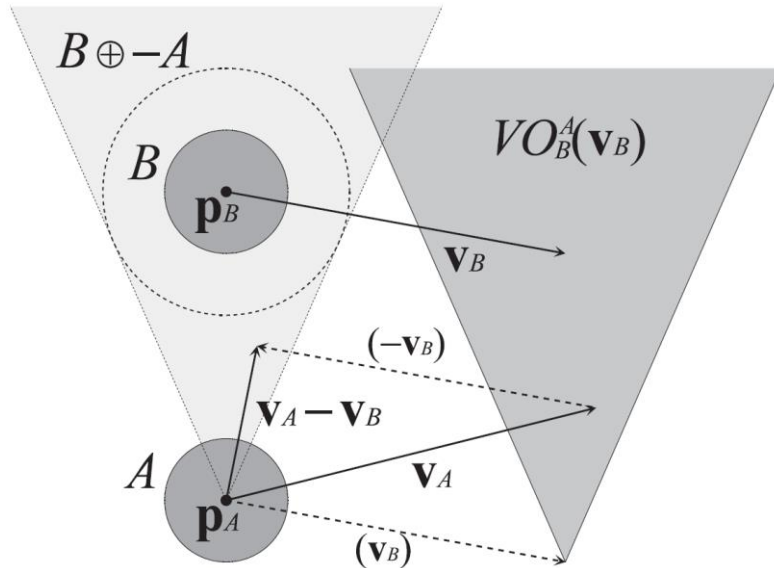
Yu ve arkadaşları [62], merkezkaç kuvveti modeli yaklaşımlarında bireyin önünde giden hızlı bireylerin bireye itme kuvveti uygulamasını önlemenin yanı sıra saldırgan bireylerin

öndeki boşluklara doğru ilerlemesini de göz önünde bulundurmuşlardır. Boşaltma senaryolarında kapı genişliğinin birey çapının beş katından büyük olması durumunda kapıda yığılma olasılığının sıfıra yaklaştığını belirlemişlerdir. Ayrıca boşaltma hızının kapı genişliğiyle arasındaki ters orantı katsayısını hesaplamışlardır.

2.3.1.2.2.3. Geometrik Tabanlı Modeller

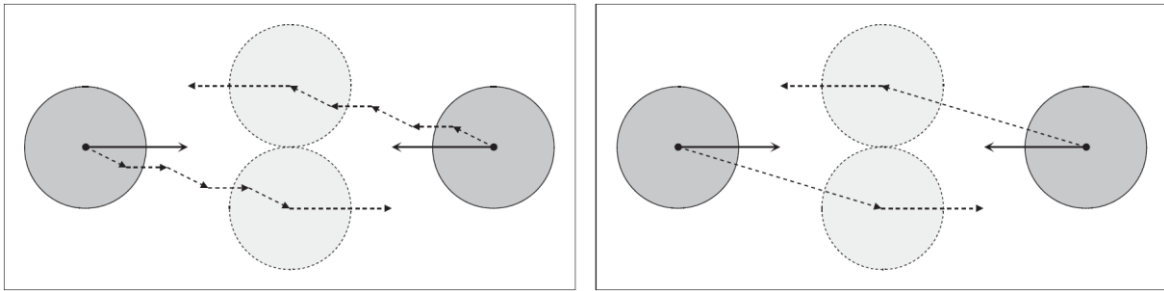
Geometrik tabanlı modellerde her bireye belirli sabit bir süreliğine çarpışmasız olduğu garanti edilen hız vektörleri tayin edilir. Gerek sabit gerek hareketli engellerle çarpışmasız olmayı garanti edebilen bu vektörlere göre hareket eden bireyler için hedef konumlarına ulaşana kadar belirli aralıklarla tekrar hız vektörü hesaplanır.

Fiorini ve Shilller, her bir birey için çevrelerindeki sabit ve hareketli engellere göre çarpışmasız hız vektörleri hesaplayan bir yöntem geliştirmişlerdir [63]. Özellikle hareketli engelleri ele alırken geliştirdikleri yaklaşım önemli olmakla birlikte bu yaklaşım doğrultusunda hareket eden bireylerin görel hızlarına ve kapladıkları dairesel alanın yarıçaplarına göre hız vektörü uzayında belirli bölgeleri çarpışma olacak kısımlara karşılık gelen hız vektörü engelleri (Şekil 2.11) olarak belirlemişlerdir. Hız vektörü uzayında çevredeki hareketli ve hareketsiz her nesnenin hız vektörü engellerini birleştirdikten sonra serbest kalan alanlar içerisinde bireyin izin verilen hız ve doğrultu değiştirme sınırları dâhilinde çarpışmasız bir hız vektörü belirlemişlerdir.



Şekil 2.11 Disk şeklinde ifade edilen A bireyinin yine disk şeklindeki B bireyinden kaynaklı hız vektörü engeli [64]

Fiorini ve Shiller tarafından geliştirilen hız vektörü engeli yöntemi bireylere çarpışmasız hız vektörleri bulmak konusunda yeterince başarılı olmasına karşın her birey hız vektörü uzayında diğer tüm bireylerin hız ve doğrultusunu değiştirmeyeceğini varsayması nedeniyle olası çarpışmalardan kaçınmak üzere tam manevra yapar. Bireyin etrafındaki diğer bireyler de aynı şekilde tam manevra yapacağı için bireylerde sürekli hız vektörü değişikliğine bağlı olarak Şekil 2.12 (Sol)'daki gibi salınımlar görülür. van den Berg ve arkadaşları [64], bu salınım problemine çözüm getirmek üzere bireyin çarpışmasız hız vektörü belirlerken tam manevra yapmak yerine çevresindeki diğer bireylerin de kendisinden kaçınacağını varsayarak kısmi manevra yapmasını sağlamışlardır (Şekil 2.12 (Sağ)).



Şekil 2.12 (Sol) Hız vektörü engeli yöntemi ile tam manevra yapılması sonucu oluşan salınımlar, (Sağ) Kısmi manevra yapılması ile salınım probleminin önlenmesi [64]

Guy ve arkadaşları [65], Fiorini ve Shiller tarafından ortaya konan hız vektörü engeli yaklaşımını ikinci dereceden en iyileştirme (optimizasyon) problemi olarak ele alıp her bireyin çarpışmasız hız vektörlerindeki değişiklikleri en aza indirmeyi amaçlayan bir yöntem geliştirmişlerdir. Ayrıca geliştirdikleri bu yöntemin veri ve işlem paralelliği özelliklerini kullanarak koştur işletim yoluyla hızlandırılmasını sağlamışlardır.

Guy ve arkadaşları [13], hız vektörü engeli yaklaşımı ile bireyler için mümkün olan çarpışmasız hız vektörlerini belirledikten sonra her bir birey için bunlar arasından enerji tüketimini en aza indirecek olanlarını seçerek bireylerin en az çaba prensibiyle [10] uygun biçimde hedeflerine mümkün olan en kısa yoldan, en az manevra ve en uygun hız değeri ile ilerlemelerini hedefleyen bir gezinim yaklaşımı önermişlerdir. Bu sayede bireylerin gerçek hayattakine benzer olarak, hedeflerine çarpışmasız bir biçimde ulaşmak üzere minimum enerjiyi harcayarak benzetimini ve bu esnada planlanmadığı halde kendiliğinden ortaya çıkan birçok davranışın da gerçekleştirilmesini sağlamışlardır.

2.3.1.2.3. Orta Ölçekli Modeller

Orta ölçekli modeller, büyük ve küçük ölçekli modellerin olumlu özelliklerini bir araya getirerek bireylerin hareket ve davranış kalitesi ile benzetim maliyetini ortalama bir düzeyde tutmaya çalışır. Orta ölçekli modeller, büyük ölçekli modellerin aksine kalabalığın geneli için değil tek tek bireyler bazında gezinim kararlarının alınmasını gerçekleştirir. Bu nedenle çoğu zaman küçük ölçekli modeller grubuna dâhil edilmektedirler. Ancak orta ölçekli modellerde bireylerin hareket ve davranışlarının gerçekçiliği küçük ölçekli modeller seviyesine ulaşamaz. Orta ölçekli modeller, gerçekleştirdikleri hareket ve davranışlar çok detaylı olmadığı için işlem gücü ve işletim süresi kazancı sağlamaları nedeniyle performans anlamında küçük ölçekli modeller ile büyük ölçekli modeller arasında yer alırlar.

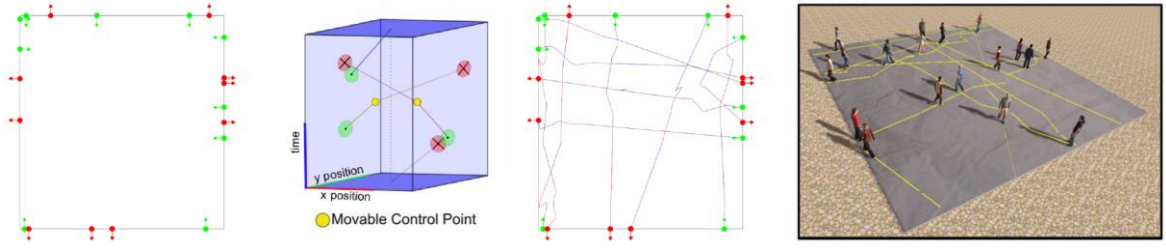
Blue ve Adler [66], benzetim alanını bireylerin tam sığabileceği büyüklükte hücrelerden oluşan düzgün bir ızgara şeklinde tanımlamış ve bireylerin birim zamanda buldukları hücreden komşu hücrelere geçmelerini sağlayacak çeşitli kurallar vasıtasıyla sabit hızda hareket etmelerini sağlamışlardır. Bu yöntemde yüksek yoğunluklu kalabalıkların gerçekçi benzetiminin yapılması mümkün olmasa da bulunulan hücreden basit kurallarla komşu sekiz hücreye geçilmesi ve birey temelli kimi basit davranışların modellenmesi mümkündür.

Benzeri bir mantıkla çalışan EGRESS [67], boşaltma senaryolarının benzetimini yapabilmek için geliştirilmiş ticari bir yazılımdır. Benzetim alanını oluşturan hücrelere çeşitli özellikler yerleştirilerek boş hücre, birey içeren hücre, duvar veya engel gibi tanımlamaların yapılması mümkündür.

Aynı zamanda birden fazla ızgara kullanarak benzetim yapan örnekler de mevcuttur. Kretz ve Schreckenberg [68], her hücrenin çıkışa olan uzaklığını saklayan bir tane statik, bireylerin hücreler arası geçiş bilgilerini saklayan bir tane dinamik ve her hücrenin engellere ve duvarlara uzaklığını tutan bir tane statik olmak üzere üç farklı ızgara kullanmışlardır.

Tüm bunların dışında Yersin ve arkadaşları [6] çok farklı bir yaklaşım olan kalabalık parçaları yönteminde Şekil 2.13'te bir örneği sunulan "parça" (*patch*) ismini verdikleri 2 boyutlu yapı taşları üzerinde bu parçaların kenarları arasında seyahat etmeyi sağlayan çarpışmasız yörüngeler tanımlamışlardır. Benzetim öncesinde çevrimdışı bir biçimde oluşturulan bu yapı taşlarının tıpkı bir yapbozun parçaları gibi yan yana gelenlerinin kenarları üzerindeki yörünge başlangıç ve bitiş noktaları birbirine denk gelerek yörüngeleri bu parçalar arasında da devam ettirecek şekilde birleştirilmesiyle çok geniş benzetim alanları oluşturmuşlardır. Parçalar üzerindeki çarpışmasız yörüngelerin periyodik olmasını, yani belirli zaman aralıkları ile bu yörünge üzerinde bir bireyin hareket etmesini sağlayarak

benzetim alanları üzerinde elde edilen bağlantılı çarpışmasız yolları izlemek suretiyle hareket eden kalabalıklar oluşturulmasını sağlamışlardır.



Şekil 2.13 Kalabalık parçaları yönteminin yapı taşları olan parçalardan birinin farklı görünüşleri [7]

Ramirez ve arkadaşları [7] bu parçalar üzerinde kullanılan çarpışmadan kaçınma yaklaşımını optimizasyon tabanlı bir yöntemden faydalanmak suretiyle geliştirerek daha yumuşak parça içi yörüngeler üretmeyi başarmışlardır. Jordao ve arkadaşları [8] kalabalık parçaları fikrini genişleterek benzetim alanını oluşturan parçaların her biri üzerinde yoğunluk ve akışın yerel olarak ayarlanabilmesine olanak sağlayan parametreler eklemiştir. Kalabalık parçaları yöntemlerinin değişik türevleri, kullanıma hazır konum-zamansal yapı taşlarından faydalanarak düşük maliyetli gerçek zamanlı kalabalık benzetimleri gerçekleştirme olanağı sunmaktadır. Ancak bu yöntemlerin hepsi, ön tanımlı parçaların oluşturulması için zaman alıcı, çevrimdışı hesaplama süreçlerine ek olarak hesaplanan parçaların saklanması için bellek tüketimi yüksek büyük veri tabanları ihtiyacını da beraberinde getirmektedir.

2.3.1.2.4. Karma Modeller

Karma modellerde büyük ölçekli ve küçük ölçekli modellerin bir arada kullanıldığını söylemek mümkündür. İki modelin birlikte kullanımını sağlamak adına çeşitli yaklaşımlar geliştirilmiştir.

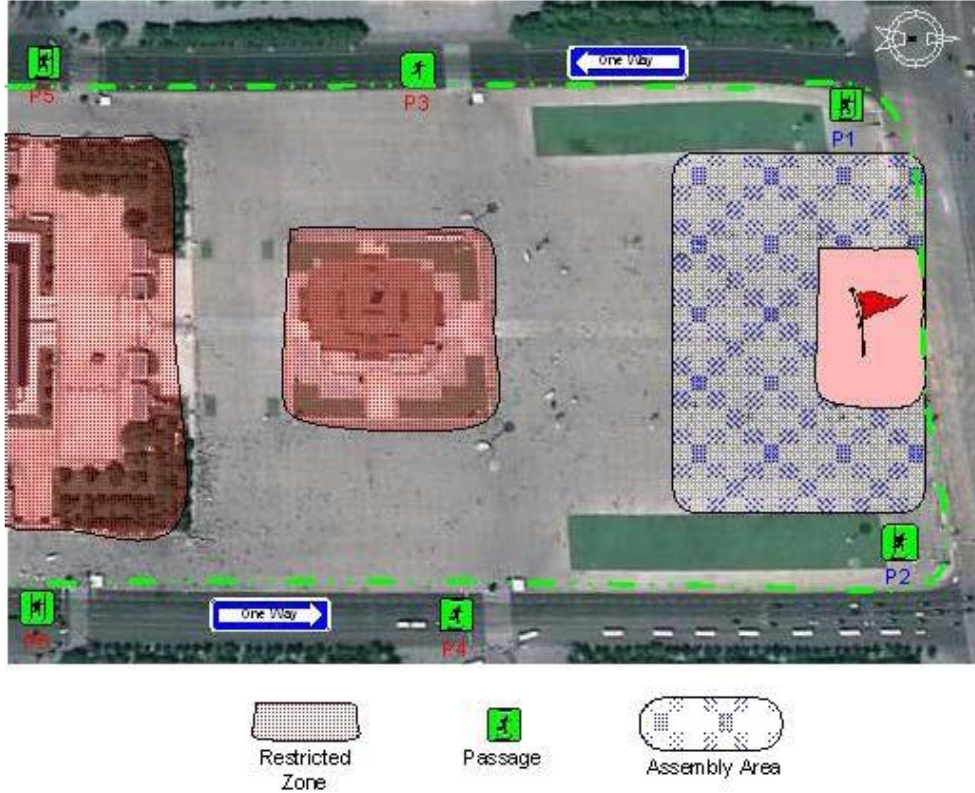
2.3.1.2.4.1. Bölge Tabanlı Modeller

Bölge tabanlı modellerde, benzetim alanı çeşitli bölgelere ayrılarak ihtiyaca göre kimi bölgelerde büyük ölçekli kimi bölgelerde ise küçük ölçekli modeller kullanılır. Bu sayede benzetimin sadece küçük ölçekli modellere göre daha verimli çalışması sağlanırken sadece büyük ölçekli modellere göre daha kaliteli bir benzetim elde edilmesi mümkün olmaktadır.

Anh ve arkadaşları [69], şehrin yolları üzerinde bireyleri hareket ettirmek için yolları kenar, yolların kesişimlerini ise düğüm olarak kabul eden bir çizge oluşturmuşlardır. Bu sayede

yolların kesişim noktaları ve yakın kısımlarında birey tabanlı küçük ölçekli bir model, geri kalan kısımlarında ise büyük ölçekli bir model kullanmışlardır.

Wei ve arkadaşları [70], Şekil 2.14'te gösterildiği şekilde bir şehir meydanında kalabalığın yoğun olduğu bölgeyi akış tabanlı bir yaklaşım ile, bireylerin hareket ettiği diğer seyrek bölgeleri ve meydanın etrafındaki araçların hareket ettiği yolları ise birey tabanlı bir yaklaşımla modellemişlerdir.



Şekil 2.14 Benzetim alanının farklı bölgelerinde ihtiyaca göre değişik modellerin kullanılması [70]

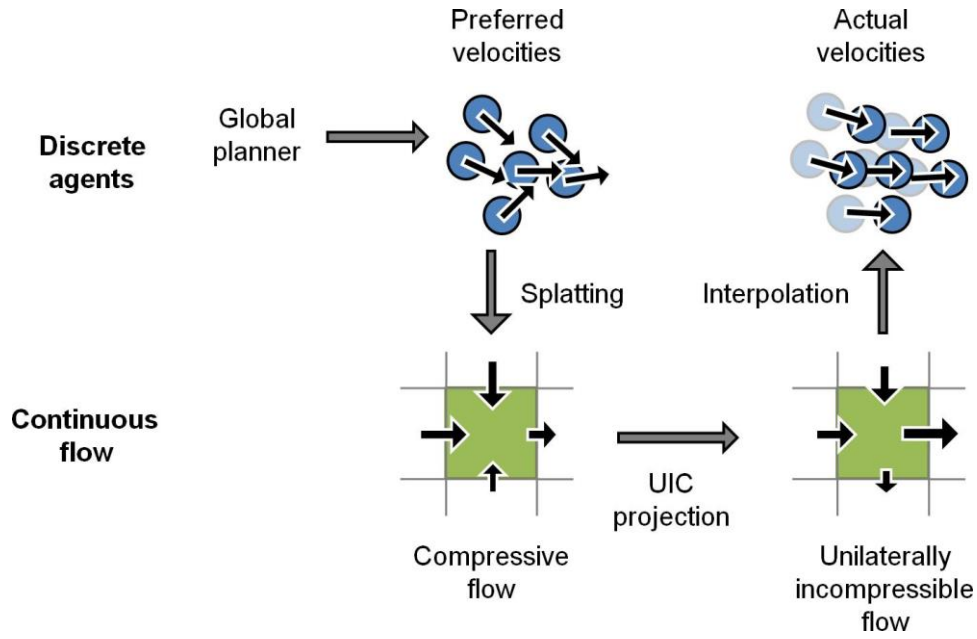
Sewall ve arkadaşları [38], bir şehirdeki araçların trafik benzetimini gerçekleştirirken kameraya yakınlık, sistemin anlık performans durumu gibi bilgilere göre dinamik olarak benzetim alanının bölgelere ayrılmasını ve kaliteli benzetime ihtiyaç duyulan bölgelerde küçük ölçekli, diğer bölgelerde ise büyük ölçekli modellerin kullanılmasını sağlamışlardır.

Xiong ve arkadaşları [71], dar bir koridor ile birbirine bağlanan iki geniş alandan oluşan benzetim ortamlarında, geniş alanlarda küçük ölçekli, dar koridorda ise büyük ölçekli bir model kullanarak benzetim gerçekleştirmişlerdir.

2.3.1.2.4.2. Sıralı Modeller

Sıralı modellerde öncelikle büyük ölçekli model çalıştırılarak kalabalığın hareketi hakkında genel bir bilgi toplanır. Sonrasında bu üst düzey bilgi küçük ölçekli modele iletilerek bireylerin çevresel bilgilerinin yanı sıra daha olan genel bu bilginin de kullanılması sağlanır. Böylelikle daha doğru bir benzetim yapılmasına yardımcı olunur.

Narain ve arkadaşları [72], akış tabanlı bir modelden elde ettikleri üst düzey bilgi ile bireylerin tercih edilen hız vektörlerinin iyileştirilmesini sağlamış, daha sonra da bireyleri bu hız vektörlerine göre hareket ettiren küçük ölçekli modeli kullanarak (Şekil 2.15) yerel çarpışma engelleme maliyetini en aza indirmişlerdir.



Şekil 2.15 Narain ve arkadaşları [72] tarafından geliştirilen karma modelin genel sistem yapısı

Haciomeroglu ve arkadaşları [73], birbirine yakın konumlarda bulunan ve birbirine yakın hız vektörlerine sahip olan bireyleri GİB üzerinde paralel bir şekilde kümelemişlerdir. Daha sonra benzetimde yer alan bireylerin bu kümelerin her birini daha büyük tek bir birey olarak ele almalarını ve kümenin içine doğru yol almayıp kümenin etrafından dolaşmalarını sağlayarak daha gerçekçi bir benzetim imkânı sunmuşlardır.

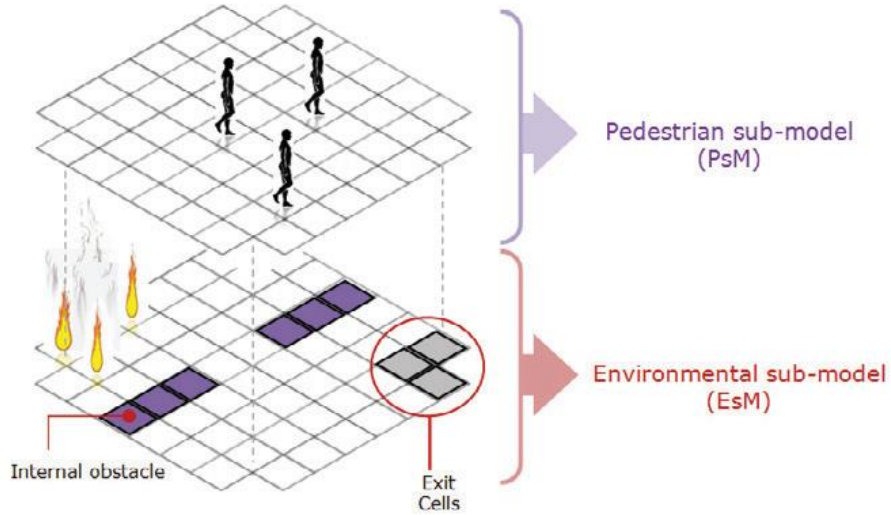
Xiong ve arkadaşları [74], büyük ölçekli akış tabanlı bir modelden aldıkları üst düzey bilgiyi hücresel özdevinir (otomat) tabanlı bir yöntemle ileterek benzetim performansını iyileştirmişlerdir.

2.3.1.2.4.3. Katman Tabanlı Modeller

Katman tabanlı modellerde üst katmanda büyük ölçekli modelin çalışması, alt katmanda da küçük ölçekli modelin çalışması sağlanır. Böylece büyük ölçekli model bireylerin genel hareketlerine karar verirken küçük ölçekli model bu hareketlerin gerçekleştirimi sırasında çarpışma tespiti ve çarpışmadan kaçınma manevraları gibi işlemleri ele alarak daha kaliteli bir benzetim elde edilmesine yardımcı olur.

Patil ve arkadaşları [75], bireyleri hedeflerine doğru götürürken üst katmanda akış tabanlı bir model kullanmakla birlikte bireylerin hareketi esnasında maruz kalacağı çarpışmaları çözmek için fizik tabanlı veya geometrik tabanlı bir model kullanmışlardır (Şekil 2.16).

Tissera ve arkadaşları [76], boşaltma senaryosu benzetimlerinde üst katmanda bireyleri çıkışa doğru yönlendiren orta ölçekli hücresel özdevinir tabanlı bir model kullanırken alt katmanda bireylerin birbirleriyle ve engellerle etkileşimlerini ele alan küçük ölçekli bir modelden faydalanmışlardır.



Şekil 2.16 Tissera ve arkadaşları [76] tarafından geliştirilen modelin katman yapısı

Haciomeroglu ve arkadaşları [77], daha önce kullandıkları kümeleme mantığını geliştirerek üst katmanda geometrik tabanlı bir yöntem ile bireylerin oluşan diğer kümeleri yine büyük birer birey olarak varsayıp bunlardan kaçınacak birer hız vektörü belirlemelerini sağlamışlardır. Alt katmanda ise bu hız vektörlerini tercih edilen hız vektörü olarak kural tabanlı, fizik tabanlı veya geometrik tabanlı küçük ölçekli bir modele girdi olarak sunmuşlardır.

2.3.2. Kalabalık Benzetiminde Görselleştirme

Görselleştirme, kalabalık benzetiminin kullanıcıya bakan yüzü olarak tanımlanabilir. Görselleştirme sayesinde kullanıcılar benzetimin neyi amaçladığı, nasıl çalıştığı, ne kadar gerçekçi olduğu, ne kadar başarılı olduğu gibi birçok soruya cevap bulabilmektedir. Bu durumun benzetim bağlamında gerçekleştirilen görselleştirmenin kalitesiyle doğru orantılı olduğunu söylemek mümkündür.

Kalabalık benzetimlerinde kaliteli bir görselleştirme için:

- Bireylerin vücut hareketlerini gerçekleştirirken vücut modellerinde göze hoş görünen yumuşak deformasyonlar oluşturabilmek adına yüksek poligon sayısına sahip modeller kullanılmalıdır [78].
- Bireylerin hareket animasyonlarını gerçekçi bir şekilde yapabilmek için uygun bir iskelet yapısı tercih edilmelidir [49].
- Fotoğraf gerçekliğinde bir görselleştirme için maliyetli ışıklandırma ve gölgelendirme işlemlerine yer verilmelidir [78].
- Bireylerin farklı duygu ve düşüncelerini ifade etmek üzere farklı yüz ifadelerinin kullanılmasına olanak tanınmalıdır [49].
- Bireylerin görünüş ve hareket animasyonu çeşitliliğinin sağlanması için gerekli tekniklerin bütünleştirilmesi sağlanmalıdır [53].
- Bireylerin saçlarının ve giysilerini oluşturan kumaşların gerçek hayattaki fizik kurallarına uygun hareketini sağlayacak yöntemler kullanılmalıdır [49].

Tez kapsamında kalabalıkların görselleştirilmesinden çok gezinim problemiyle ilgilenildiği üzere kalabalık benzetiminde görselleştirme konusunun derinlemesine incelenmesine ve konu ile ilgili çalışmaların örneklenmesine gerek duyulmamıştır. Yine de kalabalık benzetiminde görselleştirme konusuna ilgi duyanlar için daha detaylı bilgi içeren kapsamlı alanyazın taramaları [49], [53], [78], [79] mevcuttur.

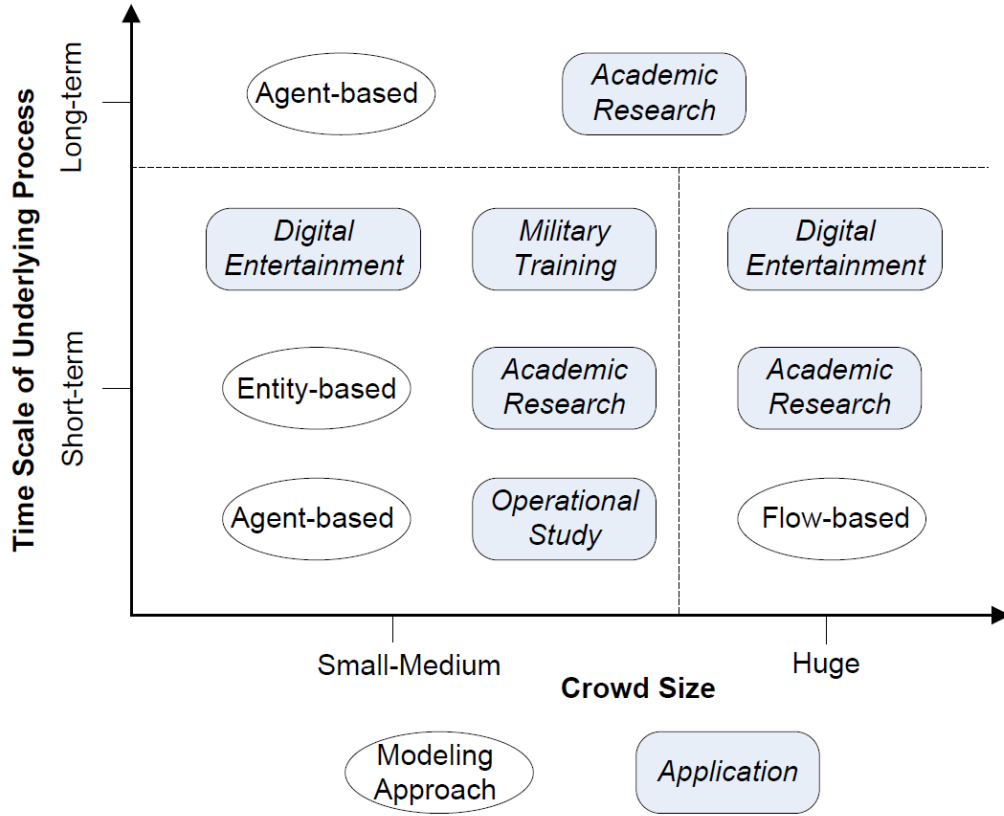
2.4. Kalabalık Benzetiminin Kısıtlamaları

Kalabalık benzetiminin doğası gereği birey sayısı çoğu zaman en önemli unsurdur. Benzetimin kullanım alanına ve amacına uygun olarak yeterli sayıda bireyden oluşan bir kalabalık oluşturulması gerekir. Ancak herhangi bir sayıda birey içeren her kalabalığın en kaliteli ve/veya en gerçekçi şekilde benzetiminin yapılması gerek davranış ve hareket modellenmesi gerekse görselleştirme alanında bilinen en başarılı yöntemlerin kullanılmasıyla

dahi mümkün olmamaktadır. Bir diğer deyişle var olan en etkili yöntemler bile benzetimi yapılabilecek maksimum birey sayısı için bir sınırlama oluşturmaktadır.

Bu noktada yapılması gereken, benzetimi yapılacak birey sayısı belirlendikten sonra benzetimi oluşturan bireylerin ne kadar kaliteli ve/veya gerçekçi bir davranış ve hareket modellemesi ile görselleştirmeye ihtiyacı olduğu konusunu gündeme almaktır. Çünkü gerek davranış ve hareket modellerinin gerekse görselleştirme tekniklerinin işlem gücü maliyetlerinin yanı sıra diğer kaynak gereksinimlerinin farklılık göstermesi söz konusudur. Bu sayede ihtiyaca göre var olan bu model ve tekniklerin en uygun olanları seçilerek benzetim için kullanılmaktadır. Ancak genel olarak maliyeti düşürerek daha fazla bireyin benzetiminin yapıldığı durumlarda benzetim kalitesinin ve/veya gerçekçiliğinin azaldığı bilinen bir gerçektir.

Zhou ve arkadaşları [16], kalabalığın boyutlarını onlarca bireyden oluşanlar için “küçük boyutlu”, yüzlerce bireyden oluşanlar için “orta boyutlu”, binlerce bireyden oluşanlar için “büyük boyutlu” olmak üzere üç ana başlık altında sınıflandırmıştır. Ayrıca Şekil 2.17’de gösterildiği üzere kalabalığın boyutunun kullanılabilir yöntemler için belirleyici olabileceğini belirtmişlerdir. Buna göre, büyük boyutlu kalabalıklar için kalabalığı bir bütün olarak ele alarak maliyeti düşüren modellerin, küçük-orta ölçekli kalabalıklar için ise daha detaylı benzetimler sunan birey tabanlı yaklaşımların tercih edilebileceğinden bahsetmişlerdir.



Şekil 2.17 Farklı kalabalık modelleri ve uygulamalarının zaman ölçeği ve büyüklük açısından sınıflandırması [16]

2.5. Kalabalık Benzetiminde Detay Seviyesi

Daha önce belirtildiği üzere kalabalık benzetiminde kullanılan yöntemlerin kaynak gereksinimlerinin, birey sayısı ile orantılı olması nedeniyle tüm teknikler benzetimi yapılabilecek maksimum birey sayısına bir üst sınır oluşturmaktadır. Bu durum ortak bir benzetimin farklı kaynak kapasitesine sahip donanımlar üzerinde çalıştırılmasını da imkânsız hale getirmektedir. Her ne kadar benzetimde yer alan birey sayısını alternatiflerine oranla önemli ölçüde artırabilen yeni hareket ve davranış modellerinin yanı sıra yeni görselleştirme teknikleri geliştirilmiş olsa bile bu yöntemler de bir noktadan sonra benzetimin kalitesinden ve/veya gerçekçiliğinden ödün vermek zorunda kalabilmektedirler.

Bu zorunluluk sınırını daha ileriye taşımak üzere kalabalık benzetiminde detay seviyesi teknikleri kullanılmaya başlanmıştır. Bu bağlamda kullanılan detay seviyesi tekniklerinde amaç kullanıcıların ilgi ve/veya algı alanı dışında kalan bireyler için uygulanan benzetimin detay seviyesini (kalitesi ve/veya gerçekçiliğini) düşürerek performans kazancı sağlamaktır. Kullanıcıların ilgi ve/veya algı alanlarını belirtmek üzere genellikle görselleştirme amacıyla

sahneye yerleştirilmiş sanal kameraya olan uzaklık, açı gibi faktörlere ek olarak modelin hareket hızı, görselleştirme sonrası ekran üzerinde bulunacağı konum vb. parametreler de kullanılmaktadır. Kalabalık benzetiminde hem görselleştirme hem de davranış ve hareket modelleme alanında detay seviye uygulamaları mevcuttur.

2.5.1. Görselleştirme Detay Seviyesi

Detay seviyesi tekniklerinin ilk uygulamaları, arzu edilen görsel kalite seviyesine ulaşmanın çok maliyetli olduğu, bu nedenle GİB olarak bilinen özelleşmiş hızlandırıcı donanımların üretilmesine vesile olan görselleştirme alanında ortaya çıkmıştır. Bu teknikleri poligon örgüsü deformasyonu ve impostor isimli iki ana başlık altında incelemek mümkündür.

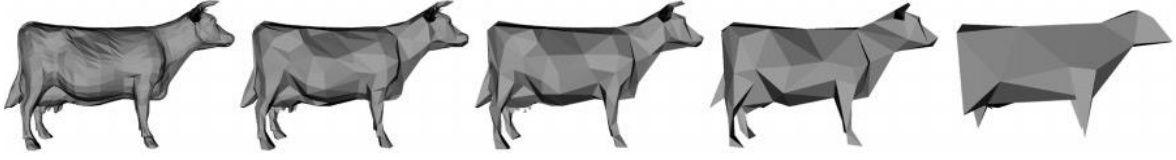
2.5.1.1. Poligon Örgüsü Deformasyonu

Poligon örgüleri, görselleştirilecek modelleri saklamaya ve işlemeye en iyi olanak tanıyan model tanımlama biçimidir. Bu yöntemde sadece örgüyü oluşturan poligonların köşelerine ait konum, yüzey normali, renk, doku koordinatı gibi bilgiler saklanmakta ve gerektiğinde modelin konum, yönelim ve büyüklüğünün değiştirilmesini sağlamak üzere gayet hızlı bir biçimde işlenmektedir. Günümüzde kullanılan GİB'ler de bu model biçimini işlemek üzere üretilmiş özelleşmiş donanımlardır.

Görselleştirilecek modelin poligon örgüsündeki poligon sayısı arttıkça modelin hatları daha yumuşak ve gerçeğe daha yakın olur. Özellikle hareketli uzuvlardan oluşan modellerde bu durum modelin animasyonu sırasında model üzerinde daha gerçekçi deformasyonların oluşmasına yardımcı olur. Tüm bu gerekliler uyarınca kalabalık benzetimlerinde bireyleri ifade etmek üzere poligon sayısı olabildiğince yüksek modeller kullanılmaya çalışılır. Ancak daha önce sözü edilen nedenlerden dolayı darboğaz oluşması kaçınılmazdır.

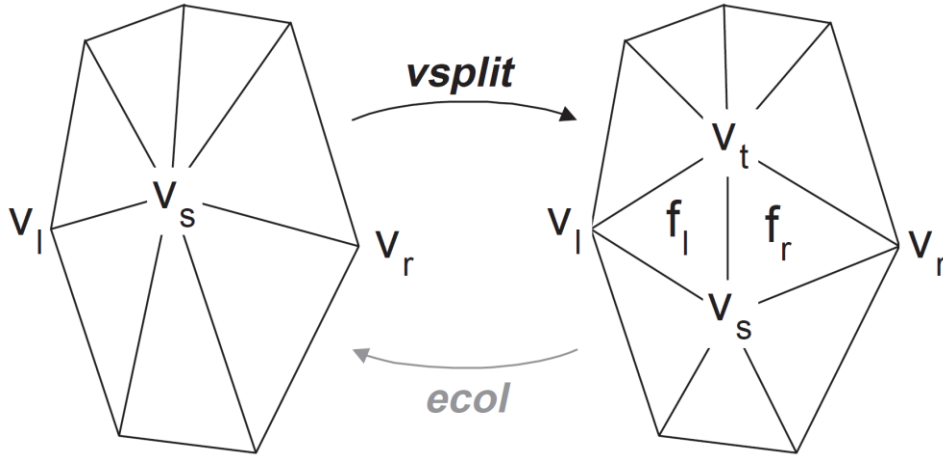
2.5.1.1.1. Poligon Azaltma

Poligon azaltma yönteminde, kullanıcının ilgi ve/veya algı alanının dışındaki bireylerin daha düşük sayıda poligondan oluşan modeller ile görselleştirilmesi esastır. Clark [80], el yordamıyla poligon sayısı azaltılmış modelleri kullanmanın görselleştirme için performans kazancı sağlayacağını fark etmiştir. Ancak elle değişik poligon çözünürlüğüne sahip modeller oluşturmak oldukça zaman alıcı bir iştir. Garland ve Heckbert, poligon azaltma işini otomatikleştirecek bir yöntem önermişlerdir [81]. Çevrimdışı bir ön adım olarak uygulanabilecek bu yöntem ile Şekil 2.18'de gösterildiği üzere modellerin poligon sayılarının istenilen seviyeye düşürülmesi ve farklı çözünürlükte oluşturulan bu statik modellerin görselleştirme için kullanılması mümkündür.



Şekil 2.18 Poligon sayısı farklı oranlarda azaltılmış 3 boyutlu model [81]

Statik yöntemlerin aksine Hoppe, “ilerlemeli poligon örgüleri” ismini verdiği dinamik poligon azaltma yöntemini geliştirmiştir [82]–[84]. Özelleşmiş bir model biçimi kullanan bu teknikte çevrimiçi olarak ihtiyaç duyulan detay seviyesine göre poligon sayısının Şekil 2.19’deki gibi artırılarak veya azaltılarak ayarlanması olanaklıdır. Sander ve arkadaşları [85], ilerlemeli poligon örgüleri üzerinde deformasyon kaynaklı kayma miktarı kabul edilebilir doku kaplamaları yapmanın mümkün olduğunu göstermişlerdir. Dinamik poligon azaltma yöntemleri, işlem maliyetleri yüksek olduğu için ancak görselleştirme sürecinden elde edilen performans kazancının bu maliyete baskın geldiği durumlarda kullanılmalıdır. Söz konusu bu maliyeti azaltmak adına Shiu ve arkadaşları tarafından yeni nesil programlanabilir GIB üzerinde çalışması öngörülen bir yöntem önerilmiştir [86].

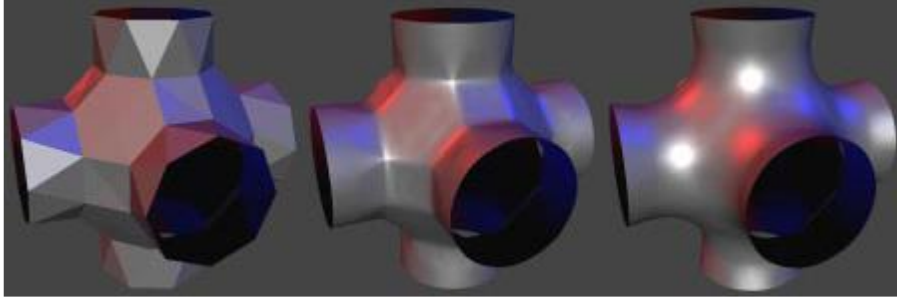


Şekil 2.19 İlerlemeli poligon örgüleri yönteminde poligon artırma (*vsplit*) ve poligon azaltma (*ecol*) işlemleri [83]

Poligon azaltma ve detay seviyesi teknikleriyle ilgili daha kapsamlı bilgi için Luebke ve arkadaşları tarafından yazılmış kitaba başvurulabilir [87].

2.5.1.1.2. Poligon Artırma

Poligon azaltma yöntemlerinin aksine zaman zaman poligon sayısı artırma ihtiyacı da oluşabilmektedir. Görselleştirme esnasında kamera bir modele yaklaştığında modelin üzerinde hareket animasyonundan kaynaklı deformasyonları daha kaliteli gösterebilmek adına modelin poligon sayısının artırılması yerinde olacaktır. Bu amaçla Loop tarafından yüzeylerin poligon sayısını yumuşak bir geçişle artırabilen bir yöntem geliştirilmiştir [88]. Zorin ve arkadaşları [89], geleneksel “kelebek” yöntemini temel alan ve topolojik hataları azaltan bir teknik önermişlerdir (Şekil 2.20).

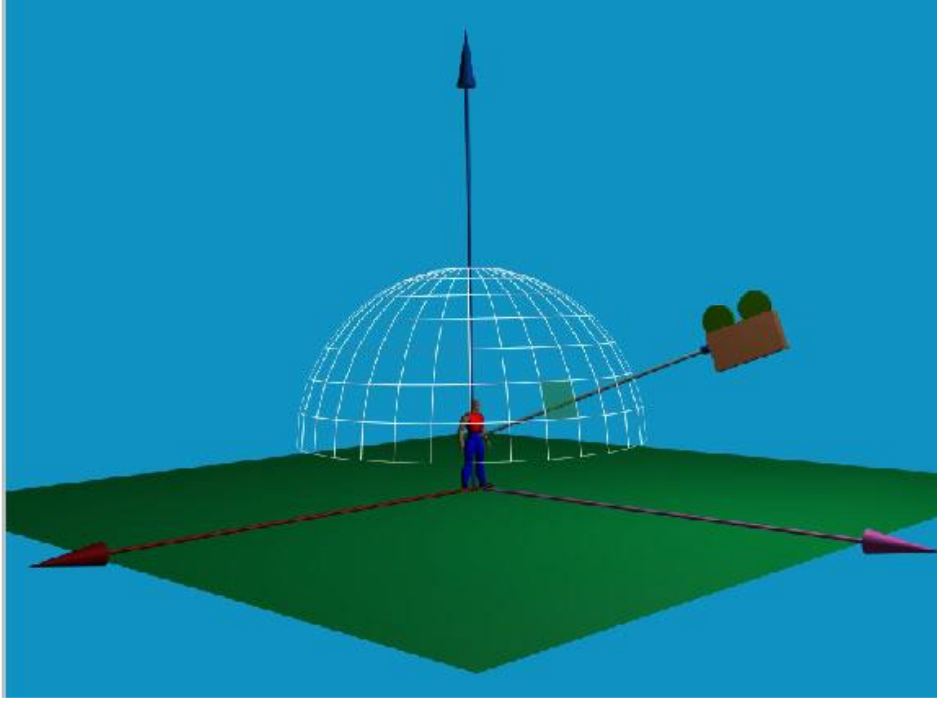


Şekil 2.20 Poligon sayısı kelebek yöntemiyle artırılan 3 boyutlu model [89]

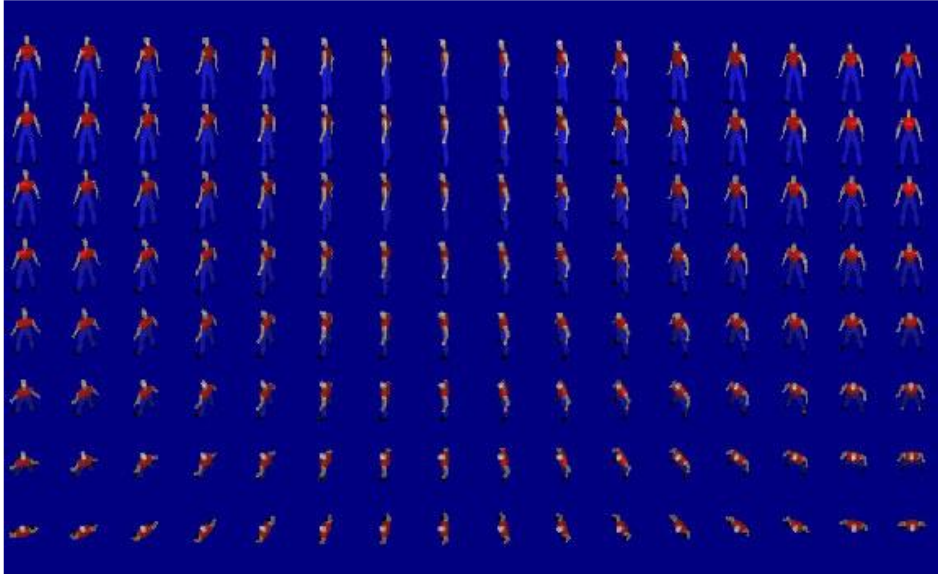
Poligon örgüsü modellerin fizik hesaplamalarını yapmak çok maliyetli olduğu için Kähler ve arkadaşları fizik hesaplamaları yapmak üzere seyrek poligonlu bir modelden faydalanırken bu modelin dinamik olarak poligon sayısını artırarak görselleştirme için kullanmışlardır [90]. Poligon sayısının dinamik olarak artırılması işlem maliyeti yüksek bir süreç olarak göze çarpar. Vlachos ve arkadaşları bu işlemleri tamamen GIB donanımı üzerinde yaparak hem işlem maliyetini başka bir kaynağa kaydıran hem de görselleştirme aşamasında GIB’ye gönderilecek model verisinin büyüklüğünü azaltmak suretiyle performans artışı sağlayan bir çalışma yapmışlardır [91].

2.5.1.2. Impostor Kullanımı

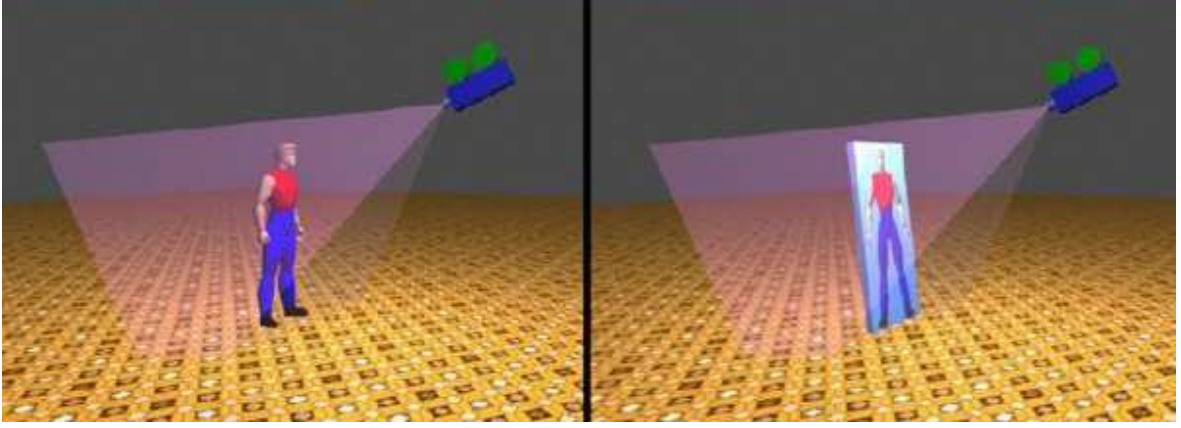
Bu yöntemde kalabalığı oluşturan bireylerin geometrik modeller olan poligon örgüleriyle görselleştirilmesi yerine bu modellerin her bir animasyon pozunu için farklı açılardan resimleri çekilir (Şekil 2.21) ve kaydedilir (Şekil 2.22). Sonrasında duruma göre uygun açılardan çekilmiş 2 boyutlu resimlerinin birer dikdörtgen üzerine kaplanması ve tabela görevi gören bu dikdörtgenlerin sürekli kameraya bakacak şekilde konumlandırılarak (Şekil 2.23) geometrik modeller ile aynı kalabalık algısının oluşturulması hedeflenir. Geometrik modellerin yerine kullanılan bu resimlere “impostor” adı verilmektedir.



Şekil 2.21 Her bir animasyon pozunu için farklı açılardan resim çekilmesi [92]



Şekil 2.22 Bir animasyon pozunu için farklı açılardan çekilerek kaydedilmiş resimler [92]



Şekil 2.23 Animasyon pozuna ve kamera açısına uygun olarak çekilmiş resmin sürekli kameraya bakan dikdörtgen tabelaya kaplanması [93]

Tecchia ve Chrysanthou [92], [93], bir ön işlem adımında bireylerin her animasyon pozuna için belirli kamera açılarından statik resimler çekerek bunları kaydetmiş ve benzetim içerisinde kullanıcı ilgisinin ve/veya algısının dışında kalan bireyler için geometrik modeller yerine bu resimleri impostor olarak kullanmışlardır. Aubel ve arkadaşları, bu resimleri önceden statik olarak hazırlamak yerine dinamik olarak benzetim içerisinde oluşturulan bir sistem geliştirmişlerdir [94]. Bu sayede impostor resimlerinin saklanması için gereken bellek ihtiyacı azalmasına rağmen dinamik olarak oluşturulan resimler kamera açısındaki değişimlere bağlı olarak kısa süreliğine geçerli olduğu için sürekli impostor oluşturulması ek işlem maliyeti gerektirmektedir.

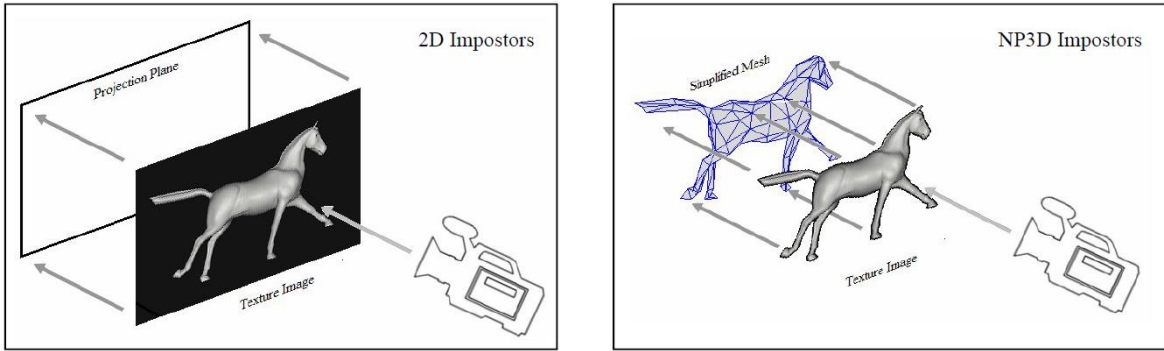
Dobbyn ve arkadaşları [1], statik impostor oluşturulması için birden fazla geçiş gerektiren önceki yöntemlerin aksine programlanabilir GİB donanımından faydalanarak çok geçiş gereksinimini ortadan kaldırmışlardır. Aynı zamanda bireylerin farklı bölgelerinin farklı renklendirilmesine ve ışıklandırılmasına olanak sağlayacak doku haritaları kullanarak hem benzetimde yer alan birey çeşitliliğinin dinamik olarak artırılmasını hem de düz bir resimden ibaret olan bireylerin yuvarlak hatlara sahipmişçesine gerçekçi ışıklandırılmasını sağlamışlardır.

Kavan ve arkadaşları [95], değişik kamera açılarından her bir animasyon pozuna için bir impostor oluşturmak yerine her kamera açısı için bireyi ifade eden modelin farklı uzuvlarına ait birer impostor oluşturmuşlardır. Farklı animasyon pozları için uygun olanlarını deforme edip birleştirerek (Şekil 2.24) ihtiyaç duyulan diğer resimleri elde etmek suretiyle bellek kazancı sağlamışlardır.



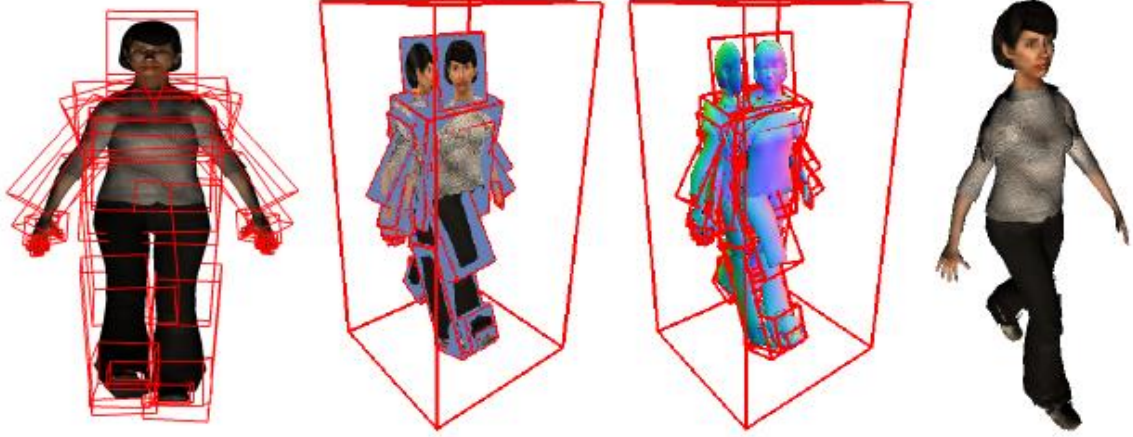
Şekil 2.24 Modelin farklı uzuvlarına ait impostor resimlerinin farklı animasyon pozları için deforme edilerek kullanılması [95]

Yee ve Davis [96], çekilen resimlerin düz bir tabelaya kaplandığı 2 boyutlu impostor kullanmak yerine çekilen resimleri modelin daha az poligon sayısına sahip düşük çözünürlüklü bir haline kaplamışlardır (Şekil 2.25). Böylelikle ihtiyaç duyulan resim sayısını azalttıkları için bellek kazancı sağlamanın yanı sıra görselleştirme sırasında impostor kullanımından kaynaklı görsel hataların da azaltılmasına yardımcı olmuşlardır.



Şekil 2.25 (Sol) Çekilen resimlerin düz bir tabelaya kaplandığı 2 boyutlu impostor kullanmak yerine (Sağ) çekilen resimlerin modelin daha az poligon sayısına sahip düşük çözünürlüklü bir haline kaplanması [96]

Beacco ve arkadaşları [97], bireylerin hareketli uzuvlarının her birini bir sınırlayıcı kutu içerisine almış ve kutunun her yüzüne ilgili uzvun o açıdan çekilmiş renk, normal ve derinlik bilgisi içeren resimlerini kaplamışlardır (Şekil 2.26). Resimleri birer kabartma gibi kullanarak her kamera açısı ve animasyon pozu için resim çekmeye gerek kalmadan impostor uygulaması gerçekleştirmişlerdir.



Şekil 2.26 Bireylerin hareketli uzuvlarının her birini bir sınırlayıcı kutu içerisine alarak bu kutuların her yüzüne ilgili uzvun o açıdan çekilmiş renk, normal ve derinlik bilgisi içeren resimlerinin kaplanması [97]

Yuksel ve arkadaşları [98], belirli açılardan az sayıda animasyon pozunu için çekilmiş resimlerin yanı sıra bu resimler üzerinde değiştirme yapmak suretiyle ara animasyon pozlarını elde ederek impostor olarak kullanmayı olanaklı hale getiren bir yaklaşım geliştirmişlerdir.

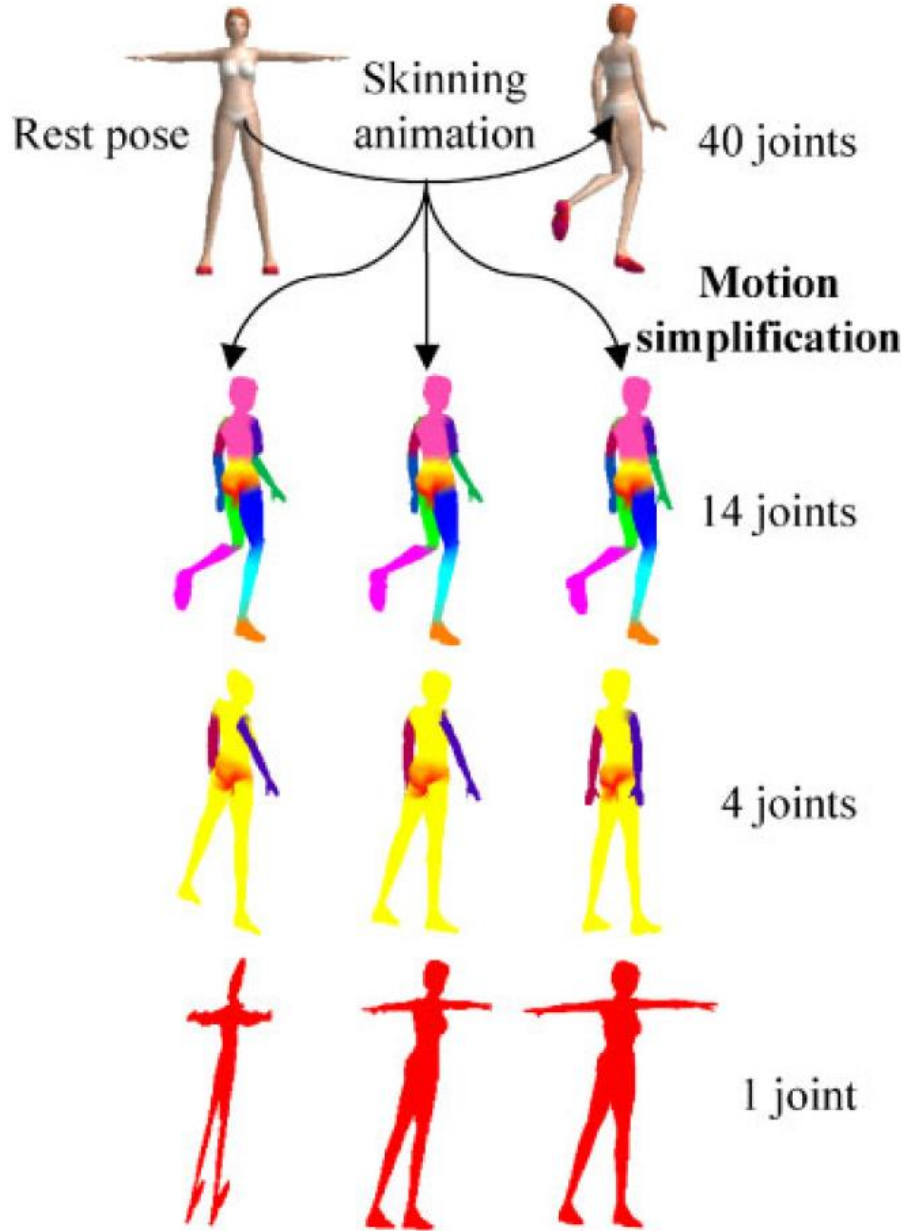
2.5.2. Davranış Detay Seviyesi

Kalabalık benzetimlerinde kullanılan davranış modellemeleri de maliyetli işlemler gerçekleştirdiği için performansı olumsuz etkilemektedir. Kalabalık benzetimlerinin performansını artırmak üzere davranış modellemesi alanında da detay seviyesi uygulamaları yapmak mümkündür.

2.5.2.1. Animasyon Detay Seviyesi

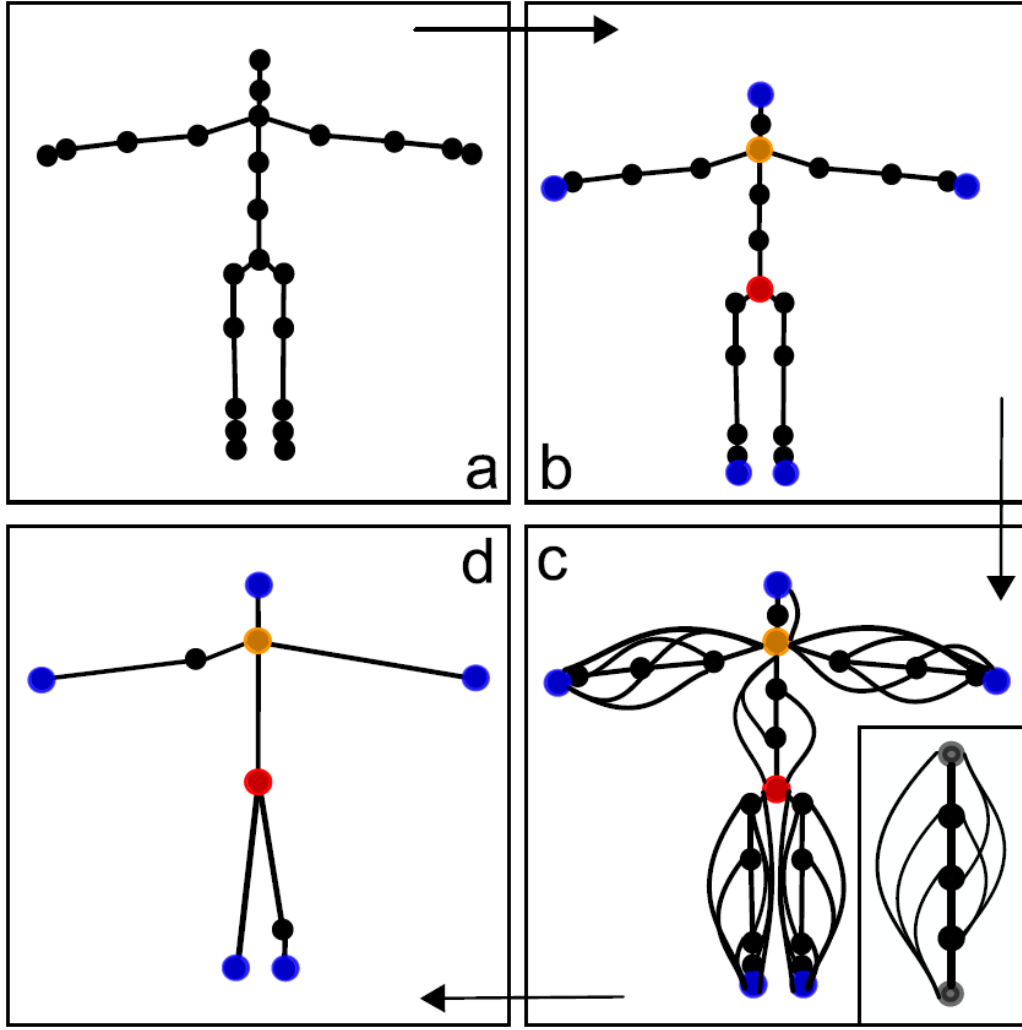
Kalabalık benzetimlerinde bireylerin hareket animasyonları benzetimin uyandırdığı gerçekçilik hissini artırmakla birlikte benzetimde kullanılan animasyon sayısı ve çeşitliliğinin artması performans sorunlarına neden olabilmektedir.

Ahn ve arkadaşları, bireylerin animasyonu için kullanılan hareketli eklem sayısını; kameradan uzaklığa, hareket hızına ve ekranın orta noktasından uzaklığa bağlı olarak azaltmak suretiyle kullanıcılara fark ettirmeden, benzetimde yer alan kimi bireylerin animasyon kalitesini düşürerek (Şekil 2.27) performans kazancı sağlayan bir sistem sunmuşlardır [2].



Şekil 2.27 Hareketli eklem sayısının azaltılarak animasyon kalitesinin düşürülmesi [2]

Savoye ve Meyer [99], bireylerin hareketli eklem sayısını değiştirmenin (Şekil 2.28) yanı sıra kullanılan animasyonlardaki kimi pozları ekleyip çıkararak kalabalık benzetiminde animasyon için kapsamlı bir detay seviyesi sistemi oluşturmuşlardır.

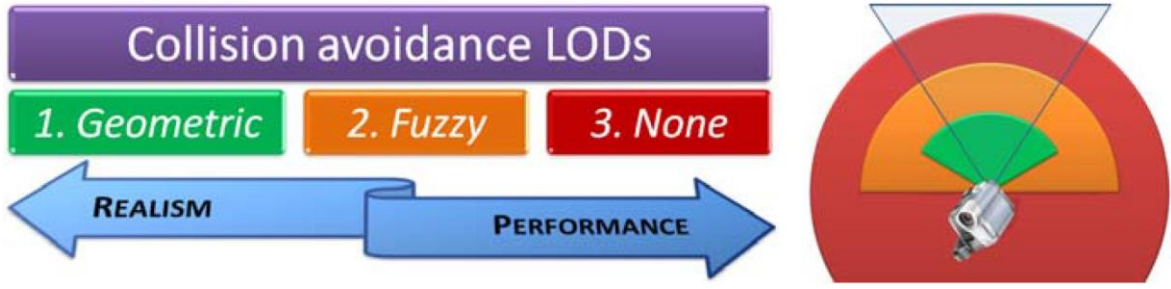


Şekil 2.28 Hareketli eklem sayısının değiştirilerek iskelet yapısının basitleştirilmesi [99]

2.5.2.2. Çarpışma Tespiti ve Engellenmesi için Detay Seviyesi

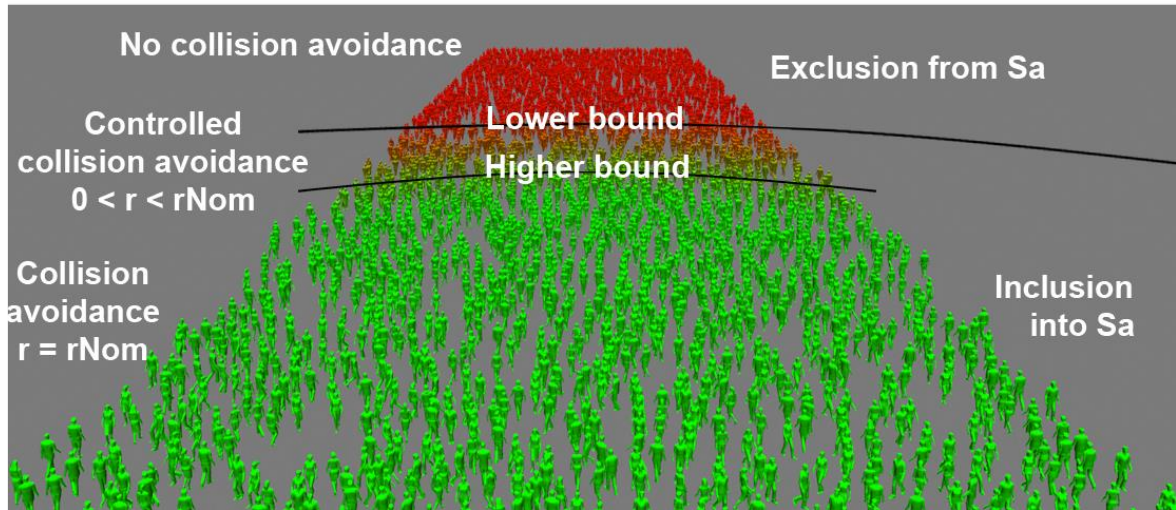
Çarpışma tespiti ve olası çarpışmaların engellenmesi, kalabalık benzetiminde bireylerin gezinim hareketlerine yön veren ve kalabalık benzetimi açısından olmazsa olmaz bir unsur olarak sayılabilecek bir mekanizmadır. Ancak olası çarpışmaların yeterince önceden tespit edilmesi ve bu çarpışmalardan kaçınmak üzere gerçekçi manevralar yapılması benzetimin kalitesini olumlu yönde etkilediği kadar performansını da olumsuz yönde etkilemektedir.

Paris ve arkadaşları, çarpışmalardan kaçınmak üzere farklı detay seviyelerinden (Şekil 2.29) meydana gelen bir sistem oluşturmuşlardır [4]. En yüksek detay seviyesinde çarpışma tespiti ve engellenmesi için geometrik tabanlı bir yöntem kullanırken orta detay seviyesinde bulanık mantık temelli bir yaklaşım geliştirmişlerdir. En düşük detay seviyesinde ise çarpışma tespiti yapılmadığı gibi çarpışmalardan kaçınılması da söz konusu değildir.



Şekil 2.29 Kameranın görüş alanının farklı bölümlerine uygulanan üç farklı detay seviyesi [4]

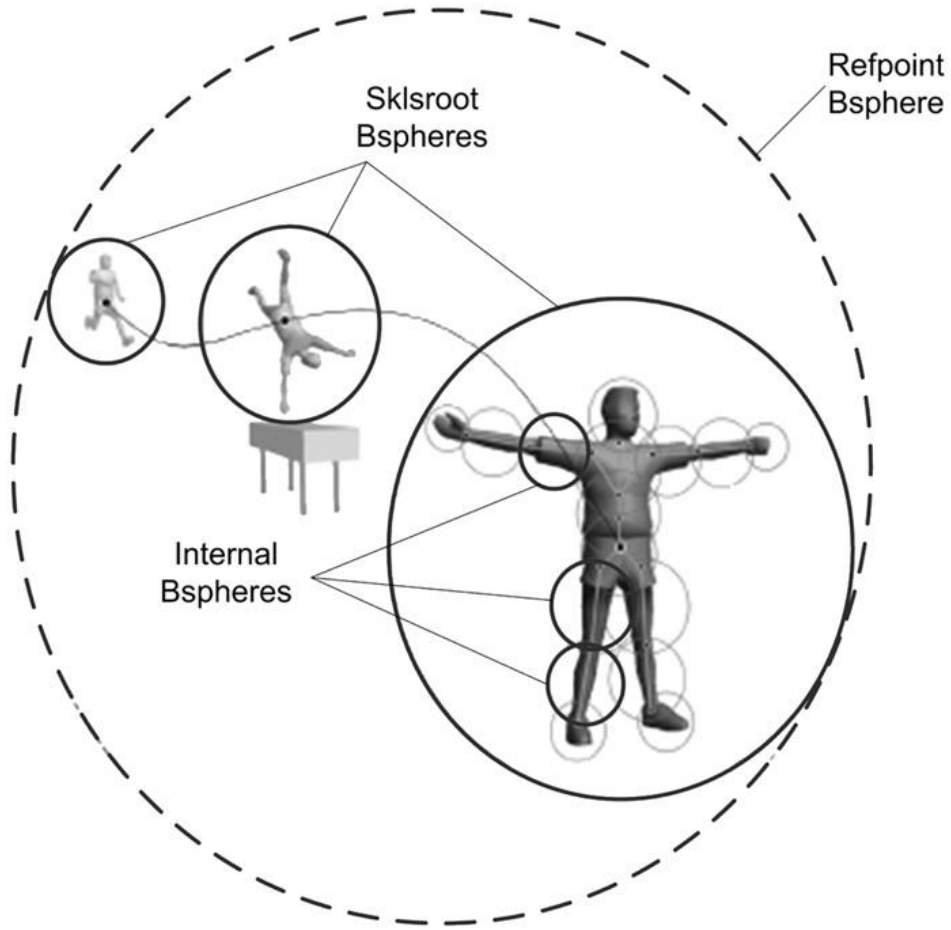
Kulpa ve arkadaşları [3], gerçekleştirdikleri kullanıcı deneyleri sonucu bireyler arası çarpışmaların kullanıcılar tarafından algılanmasında kameraya olan uzaklık, yatay ve dikey açı gibi faktörlerin etkili olduğunu ortaya koymuşlardır. Buradan hareketle bu etken faktörlere bağlı sürekli bir formül oluşturarak bireyler arası çarpışma tespiti ve engellenmesi işlemlerinde kullanılan bireylerin kapladığı alan yarıçapı bilgisini değiştirmişlerdir (Şekil 2.30). Bu sayede kullanıcılar tarafından fark edilemeyecek çarpışmalar için boş yere işlem yapılmasının önüne geçmişlerdir.



Şekil 2.30 Kalabalık benzetiminde yer alan bireylerin yarıçaplarının detay seviyesine göre ayarlanması [3]

Rodriguez ve arkadaşları [100], yüksek poligon sayısına ve hareketli uzuvlara sahip geometrik modeller ile ifade edilen bireylerden kameranın görüş alanı dışında kalanları tespit ederek bunlar için görselleştirme işlemlerini gerçekleştirilmeyip görselleştirme maliyetinden

kazanç sağlamak üzere farklı detay seviyelerinden oluşan bir model önermişlerdir. Buna göre, en düşük detay seviyesinde her bireyin hareket animasyonu boyunca üç boyutlu uzayda bulunduğu hacmi kapsayan bir küre, orta detay seviyesinde bireyin hareket animasyonu boyunca geometrik modelini kapsayan bir küre, en yüksek detay seviyesinde ise bireyin geometrik modelini oluşturan her uzvunu kapsayan birer küre kullanılmıştır. Görüş alanında bulunma testi için her bireyin en düşük detay seviyesindeki sınırları kapsayan küresinden başlayarak görülme tespit edildikçe bir üst detay seviyesindeki sınırları kapsayan küresi ile devam edilmiştir (Şekil 2.31). Bu çalışma, doğrudan çarpışma tespiti ve engellenmesi ile ilgili olmasa da benzer bir mantığa sahip olduğu için bu başlık altında incelenmiştir.



Şekil 2.31 Üç farklı detay seviyesindeki bireyler için kullanılan sınırlayıcı küreler [100]

2.6. Kalabalık Benzetimlerinde İnanırcılık

Önceki bölümlerde de bahsedildiği üzere kalabalık benzetimlerinde gerek davranış modellemesi için gerekse görselleştirme için çok farklı yöntemler geliştirilmiştir. Kullanım alanı ve amacına göre kalabalık benzetimlerinde imkânların elverdiği ölçüde en kaliteli ve

en gerçekçi benzetime ulaşılmaya çalışılır. Ancak gerektirdiği işlem gücü maliyeti göz önünde bulundurulduğunda, gerçek dünyadakine birebir benzeyen benzetimlerin gerçekleştirilmesi mümkün olmamakla birlikte detay seviye tekniklerinde başvurulduğu gibi daha çok kullanıcılar tarafından gerçekmiş gibi algılanabilen yani inandırıcı kalabalık benzetimleri tercih edilmektedir.

Kalabalık benzetimlerinde henüz inandırıcılığın net bir tanımı bulunmamakla birlikte inandırıcılık, izleyicinin algıladığı gerçeğe uygunluğun bir ölçüsü olarak kabul edilebilir. Bir başka ifade ile kullanıcının izlediği/gözlediği benzetim ile yaşadığı ve bildiği davranış biçimleri arasında fark tespit edilememesi hali inandırıcılığın başlangıç eşiği olarak kabul edilebilir. İnandırıcı kalabalık benzetiminin, yaşanan gerçeğe yüzde yüz uygun olduğunu söylemek her zaman mümkün değildir. Çünkü inandırıcılık kullanıcıyı bilfiil kendi gözlemi üzerinden ikna edebilme sürecidir ve izleyiciler arasında da önem farklılığı gösteren kıstaslara göre değerlendirilir. Kalabalık benzetimlerinde inandırıcılığı etkileyen başlıca faktörler;

- bireylerin davranış ve hareketleri
- bireylerin görünüşleri
- bireyler arasındaki etkileşim
- bireyler ve çevre arasındaki etkileşim
- bireylerin izleyiciyle/oyuncuyla etkileşimi

olarak sıralanabilir [101]. Ayrıca izleyiciler bu ölçütleri benzetimin kullanılacağı bağlama bağlı olarak da değerlendirebilirler. Örneğin, insanlardan oluşan bir kalabalık benzetiminin kullanıldığı bir filmde inandırıcı olabilmek adına gerçeğe çok yakın bir benzetim yapılması gereklidir. Çünkü insan algısı günlük hayatta sürekli karşılaştığı ve aşina olduğu insan davranış ve hareketlerindeki küçük farklılıklara bile duyarlı olacaktır. Ancak hayvanların oluşturduğu bir kalabalığın yer aldığı bir oyunda gerçek hayatla birebir örtüşmeyen bir benzetimin bile izleyiciye inandırıcı gelebileceği tahmin edilmektedir.

Benzer şekilde, çizgi film tarzında görüntü kalitesine sahip bir uygulamada insan topluluğunun benzetimi inandırıcılık açısından izleyiciye hiç rahatsızlık vermezken aynı benzetim modeli fotoğraf gerçekliğinde görüntü kalitesine sahip bir filmdeki insanlar üzerinde kullanıldığında adeta gözü tırmalayabilir. Buna ek olarak, aynı benzetim modeli aynı filmde izleyicinin ana odağından uzakta (arka planda) yer alan bir insan topluluğu

biçiminde ise inandırıcılık eşiği daha düşük olacaktır. Bu nedenle inandırıcılığı etkileyen faktörlere izleyicinin odak unsuru, görüntüleme ortamı, sunum biçimi gibi farklı etmenler de eklenebilir. Kalabalık benzetimlerinde detay seviyesi ile ilgili bölümde de bahsedildiği üzere odak faktöründen faydalanarak inandırıcılığı etkilemeyecek şekilde kalabalığın görüntü [1] veya hareket [3] kalitesini azaltmak suretiyle performans artışı sağlayıp, çok daha fazla birey içeren kalabalık benzetimleri elde edilmesini mümkün kılan çalışmalar gerçekleştirilmiştir.

Bahsedildiği gibi kalabalığı oluşturan bireylerin hareket ve davranışlarındaki çeşitlilik inandırıcılığı etkileyen en temel faktörlerden biridir. Örneğin, bir futbol oyunundaki tüm taraftarların tıpatıp aynı hareket ve davranışları sergilemesi tüm oyun tutkunlarının yıllardır şikâyet ettiği ve inandırıcılığı olumsuz yönde etkileyen bir husustur [102]. İnandırıcılığı bir nebze olsun artırmak adına taraftarların aynı hareketleri farklı anlarda farklı sıklıklarla yapması bile tek başına yeterli olmamakla birlikte her taraftarın farklı hareket eden bir birey olduğu hissini uyandıramamaktadır.

Kalabalığı oluşturan birey sayısı her ne kadar inandırıcılığı etkileyen faktörler arasında gösterilmese de [101] kalabalık algısının yaratılması için olmazsa olmaz unsurlardan biri olduğu düşünülmektedir. Örneğin, savaş meydanında bulunan insanların oluşturduğu bir kalabalığın yoğunluğu yeterli düzeyde olmadığında topluluğu oluşturan bireyler bir ordu gibi görünmekten ziyade izleyiciler tarafından bireysel hareket eden insanlar olarak algılanacak ve bu durum savaş sahnesinin inandırıcılığını önemli ölçüde azaltacaktır. Ancak sahnedeki insan yoğunluğu yeterli düzeyde ise kalabalığı oluşturan insanlar bir bütün olarak algılanıp savaştan bir ordu hissiyatını uyandıracaktır.

Kalabalık yoğunluğu ve hareket çeşitliliği faktörleri inandırıcılığı olumlu yönde etkilemekte iken birbirlerini olumsuz etkileyebilirler. Örneğin, bir fuar alanında gezen insanlardan oluşan bir kalabalık benzetiminde insan yoğunluğu arttıkça benzetimin inandırıcılığı da artacaktır. Buna ters olarak yoğunluk arttıkça fuar alanında bulunan insanların farklı doğrultu ve hızlarda hareket edebilme olanağı kısıtlanacaktır. Hatta ve hatta yoğunluğun artması bir noktadan sonra yoğun ancak hareketsiz bir kalabalık oluşmasına neden olabilir ki böyle bir sahne fuar alanından çok bir konser alanına benzeyecektir.

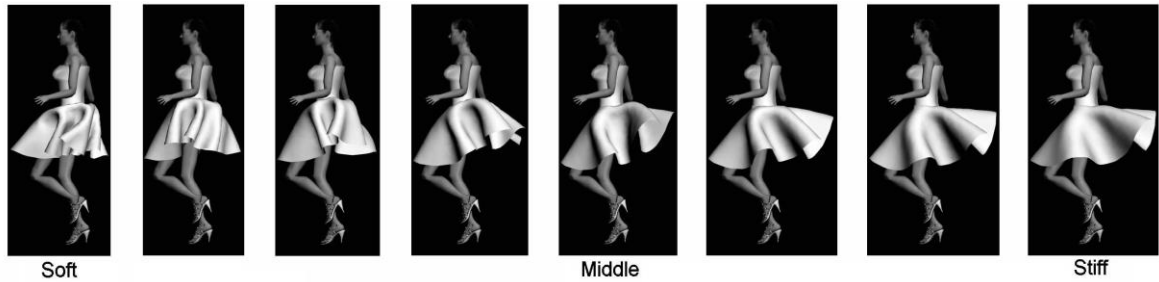
Hal böyle olunca kalabalık benzetimlerinin kalite ve/veya gerçekçiliğini kullanıcılar üzerindeki inandırıcılık etkisi açısından değerlendirmek yerinde olacaktır. Tam bu noktada devreye insan denekler üzerinde yapılan algı deneyleri girmektedir. Algı deneylerinde amaç

önerilen yöntemin ve/veya parametre değerlerinin ne kadar kaliteli ve/veya gerçekçi bir benzetim oluşturabildiğinin öznel kullanıcı değerlendirmeleri ile belirlenmesidir. Kullanıcılara değerlendirmelerinde referans oluşturabilecek bir “görece en iyi” sunulabileceği gibi tamamen kişisel tercih ve tecrübelerine göre karar vermeleri de istenebilir.

McDonnell ve arkadaşları [103], bireyleri ifade etmek üzere kullanılan farklı detay seviyesi tekniklerinin bireylerin hareket bilgisini aktarmadaki başarımını kullanıcı deneyleri ile ölçmüştür. Buna göre, impostor yönteminin hareket bilgisini aktarmada düşük çözünürlüklü model kullanımına göre daha başarılı olduğu sonucuna ulaşmışlardır.

Hamill ve arkadaşları [104], kalabalık benzetiminde impostor kullanımı sırasında yüksek çözünürlüklü geometrik modellere geçişin kullanıcılar tarafından fark edilemeyeceği en uygun parametre değerlerini belirlemek üzere kullanıcılar ile algı deneyleri gerçekleştirmişlerdir.

McDonnell ve arkadaşları [105], giysilerin kumaş benzetiminde (Şekil 2.32) kullanılan farklı detay seviyesi metotlarının başarımını değerlendirmek üzere algı deneyleri düzenlemiş ve sonuç olarak impostor yönteminin kumaş benzetiminde yüksek çözünürlüklü model kullanımı kadar başarılı olduğunu ancak düşük çözünürlüklü modellerin yeterince başarılı olmadığını görmüşlerdir.



Şekil 2.32 Giysilerin kumaş benzetiminde kullanılan farklı sertliklerdeki materyaller [105]

McDonnell ve arkadaşları, bireylerin kaliteli hareket animasyonu için kullanılması gereken minimum poz sayısını [106], model görünüşü ve/veya animasyonu aynı olan kopya bireylerin sıklığının bu kopyaların kullanıcılar tarafından tespit edilebilirliğine etkisini [107], grup halindeki bireylerin diyalogları sırasında (Şekil 2.33) hareket animasyonlarında eşzamanlılık bozukluklarının fark edilebilirliğini [108] algı deneyleri ile araştırmışlardır.



Şekil 2.33 Grup halindeki bireylerin diyaloglarını gösteren örnek bir sahne [108]

Ayrıca McDonnell ve arkadaşları, bireyleri ifade etmek için kullanılan model vücut biçimlerinin bireylerin cinsiyetlerinin [109] ve duygularının [110] algılanmasına etkisini incelemenin yanı sıra kalabalık benzetiminde birey çeşitliliğini sağlamak üzere kullanıcıların bireyler üzerinde dikkat ettiği algısal özellikleri belirleyerek bu doğrultuda birey çeşitliliğine yön veren bir yaklaşım geliştirmişlerdir [111].

Peters ve Ennis [112], kalabalık benzetimlerinde bireylerin oluşturabileceği çeşitli formasyonlar tanımlayıp bunlar arasındaki geçişleri grup üyeleri arasındaki maksimum-minimum uzaklık, bir arada kalma isteği gibi çeşitli parametrelere bağlamışlardır. Bu yöntemle farklı sayıda bireylerden oluşan grupları içeren kalabalık benzetimlerinde kullanıcı deneyleri yaparak grup sayısı ve gruptaki birey sayısı faktörlerinin benzetim gerçekçiliğine etkisini incelemişlerdir.

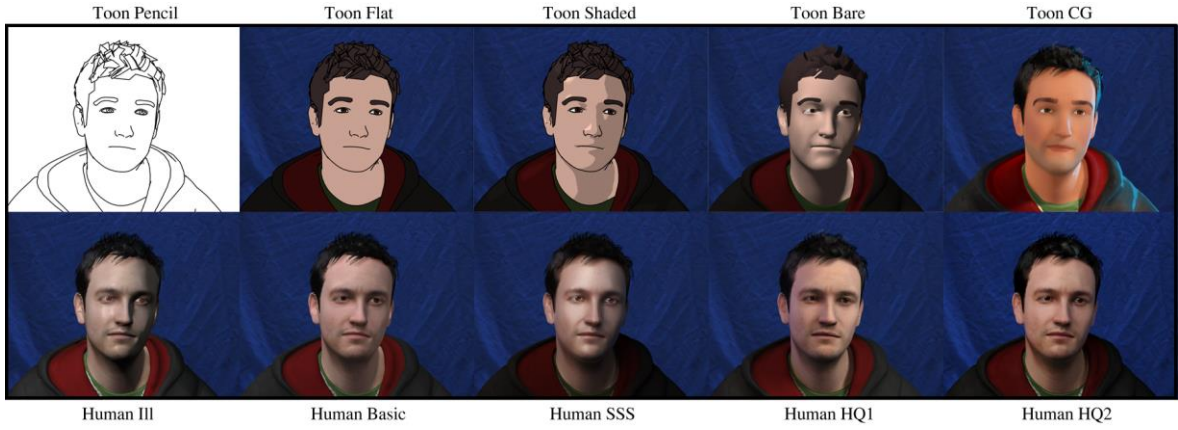
McHugh ve arkadaşları [113], kalabalığı oluşturan bireylerin farklı vücut duruşlarının, kalabalığın kullanıcı tarafından algılanan genel duygusunu nasıl etkilediğini kullanıcı deneyleri ile araştırmışlardır. Ennis ve arkadaşları, benzetim ortamına bireylerin bağlamsal kurallara göre yerleştirilmesi için geliştirdikleri sistemin geçerliliğini farklı kamera bakış açıları için algı deneyleri ile test etmişlerdir [114].

Hoyet ve arkadaşları [115], kalabalığı oluşturan bireyler arasında gerçekleşen birbirini el ile itme (Şekil 2.34) türü etkileşimlerde kuvvet miktarı, kuvvetin uygulanma açısı ve zamanlaması gibi faktörleri algılanan kaliteye etkileri açısından değerlendirmişlerdir.



Şekil 2.34 Bir karakterin diğer bir karakteri eliyle ittiği sanal bir sahne [115]

McDonnell ve arkadaşları [116], farklı görselleştirme biçimleri kullanılarak görselleştirilen bireylerin (Şekil 2.35) kullanıcılar tarafından nasıl algılandığını farklı açılardan araştırmışlardır. Jarabo ve arkadaşları, görselleştirmede kullanılan farklı ışıklandırma yöntemlerinin etkilerini incelemişlerdir [117].



Şekil 2.35 En soyuttan en gerçekçi olana doğru sıralanmış farklı görselleştirme biçimleri [116]

3. ANA İŞLEM BİRİMİ TABANLI TEK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ

Tez kapsamında ilk olarak yönlendirmesiz gezinim yaklaşımının daha düşük gezinim maliyeti sağlayıp sağlayamayacağını görmek üzere bir ön uygulanabilirlik çalışması gerçekleştirilmiştir. Bu amaçla yalnızca tek bir yörünge parçasından oluşan doğrusal ve sabit hızlı yollar kullanan yönlendirmesiz bir gezinim yöntemi önerilmiştir. Geliştirilen bu gezinim yönteminin, yeni oluşturulacak bireyler için çarpışmasız sabit hız vektörleri belirlenmesi ve bireylerin benzetim alanı üzerinde hareket ettirilmesi alt bölümlerine ayrılması mümkündür.

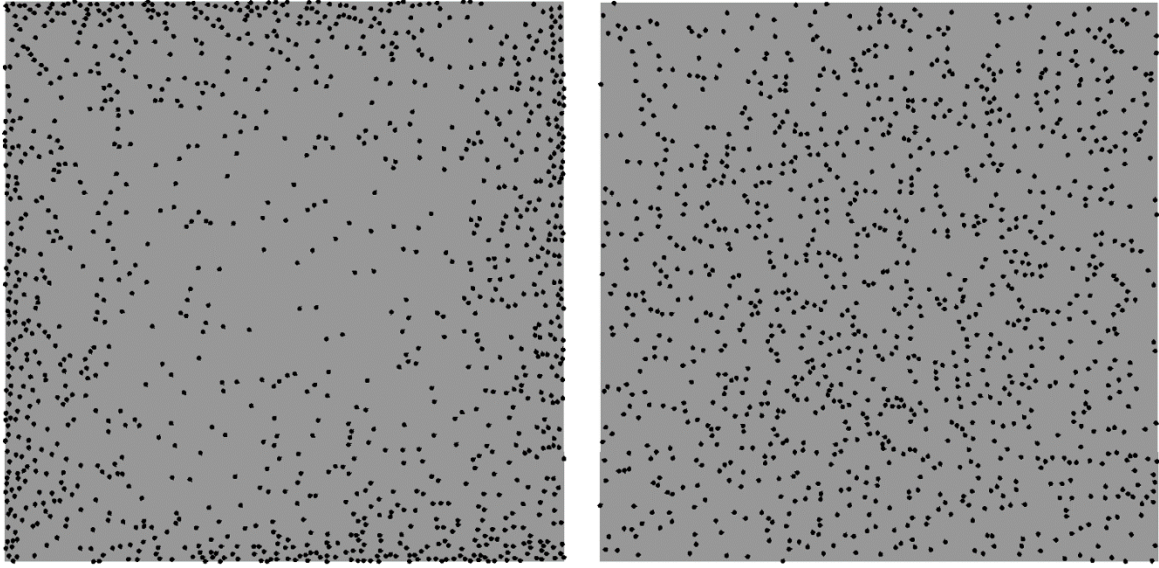
3.1. Çarpışmasız Sabit Hız Vektörleri Belirlenmesi

Bu gezinim yönteminde bireylerin hız vektörleri sabit olduğu için, yani bireylerin hareket doğrultusunu veya hızını değiştirdiği manevralar yapmalarını izin verilmediği için bireylerin kendilerine atanan yol boyunca diğer hiçbir birey ile çarpışmadığını garanti etmek gerekmektedir. Bu noktada temel problem, bir birey benzetime dâhil olmadan önce kendisi için çarpışmasız ve sabit bir hız vektörü bulunmasıdır. Bu amaçla her bir birey için yalnızca bir kere ve birey benzetime dâhil olmadan hemen önce gerçekleştirilen hız vektörü belirleme işlemi üç alt adımdan oluşmaktadır. Bunlardan ilki rastgele bir başlangıç pozisyonu ve rastgele bir hız vektörü seçilmesidir. İkincisi, benzetimde yer alan bireyler arasında bu parametreler doğrultusunda oluşturulacak yeni birey ile çarpışma ihtimali bulunanların kümesinin belirlenmesidir. Üçüncü ve sonuncusu ise yeni bireyin yörüngesi ile bu küme içerisinde yer alan bireylerin yörüngeleri arasında karşılıklı çarpışma testleri gerçekleştirilmesidir.

3.1.1. Rastgele Başlangıç Pozisyonu ve Hız Vektörü Seçilmesi

Bu ilk adımda öncelikle benzetim alanının rastgele seçilen kenarlarından biri üzerinde yer alacak rastgele bir başlangıç pozisyonu seçilir. Bitiş pozisyonu ise başlangıç pozisyonunun üzerinde bulunduğu başlangıç kenarından farklı bir kenar üzerinde yer alacak biçimde seçilir. Bitiş pozisyonunun üzerinde bulunduğu bitiş kenarının rastgele seçilen başlangıç kenarı ile komşu olduğu durumlarda benzetim alanının köşelerine çok yakın dar bir alan içerisinde nispeten daha kısa olması sebebiyle çarpışmasız olma olasılığı çok daha yüksek kestirme yollar bulunması kaçınılmaz olmaktadır. Böyle bir senaryo Şekil 3.1 (Sol)'da örneklendiği üzere benzetim alanının köşelerine yakın bölümlerinde daha yüksek birey yoğunluğuna neden olurken benzetim alanının orta bölümlerinden geçmek çarpışma ihtimalini artıracığı için bu bölümlerde oluşacak daha düşük birey yoğunlukları sonucu

benzetim alanı üzerinde düzensiz bir yoğunluk dağılımı meydana gelmektedir. Bu durumu engellemek ve bireylerin benzetim alanına daha düzenli bir biçimde dağılmasını (Şekil 3.1 (Sağ)) sağlamak üzere yeni oluşturulacak tüm bireylere başlangıç kenarının tam karşı kenarı üzerinde bitiş pozisyonu belirleme zorunluluğu getirilmiştir. Oluşturulacak yeni birey için başlangıç ve bitiş pozisyonlarına ek olarak ön tanımlı bir $[s_{min}, s_{max}]$ aralığında yer alacak biçimde rastgele bir yürüme hızı seçilir. Başlangıç pozisyonundan bitiş pozisyonuna doğru elde edilen doğrultu üzerinde seçilen bu hızla yürümeyi sağlayacak bir hız vektörü hesaplanarak olası çarpışmalar için test edilmek üzere sonraki adımlara geçilir.



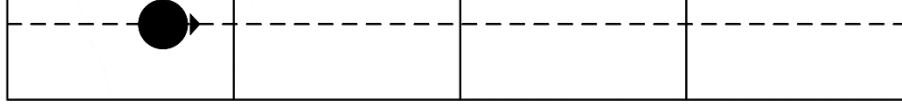
Şekil 3.1 Benzetim alanı üzerinde (Sol) düzensiz yoğunluk dağılımı ve (Sağ) düzenli yoğunluk dağılımını örnekleyen senaryolar

3.1.2. Çarpışmalara Neden Olabilecek Bireylerin Kümesinin Belirlenmesi

İkinci adımda, oluşturulan aday hız vektörünün var olan bireyler ile olası çarpışmalara karşı test edilmesi gerekmektedir. Bu amaçla kullanılacak en basit yöntem, aday hız vektörü ile benzetimde yer alan her bir bireyin hız vektörü arasında çarpışma testi gerçekleştirmektir. Ancak bu yaklaşım hiçbir zaman çarpışmanın mümkün olmayacağı bireylerle bile gereksiz yere çarpışma testleri gerçekleştirilmesine neden olacağından yeni birey oluşturmanın maliyetini yükseltmektedir. Bu nedenle benzetim alanının tamamını kapsayan ve eşit büyüklükte hücrelerden oluşan tekdüze ızgara biçiminde konum-zamansal bir dizinleme yapısı kullanılması öngörülmüştür. Şekil 3.2’de görüldüğü üzere her bir ızgara hücresi, sınırladığı alan üzerinden geçecek olan bireylerin biricik kimlik numaralarının (*id*) yanı sıra

benzetimin başlangıcına göreli olarak bireylerin bu alana giriş ve çıkış zamanlarını da tutmaktadır.

ID : 159 T _{Enter} : 12.09 T _{Exit} : 15.84	ID : 159 T _{Enter} : 15.84 T _{Exit} : 19.59	ID : 159 T _{Enter} : 19.59 T _{Exit} : 23.34	ID : 159 T _{Enter} : 23.34 T _{Exit} : 27.09
---	---	---	---

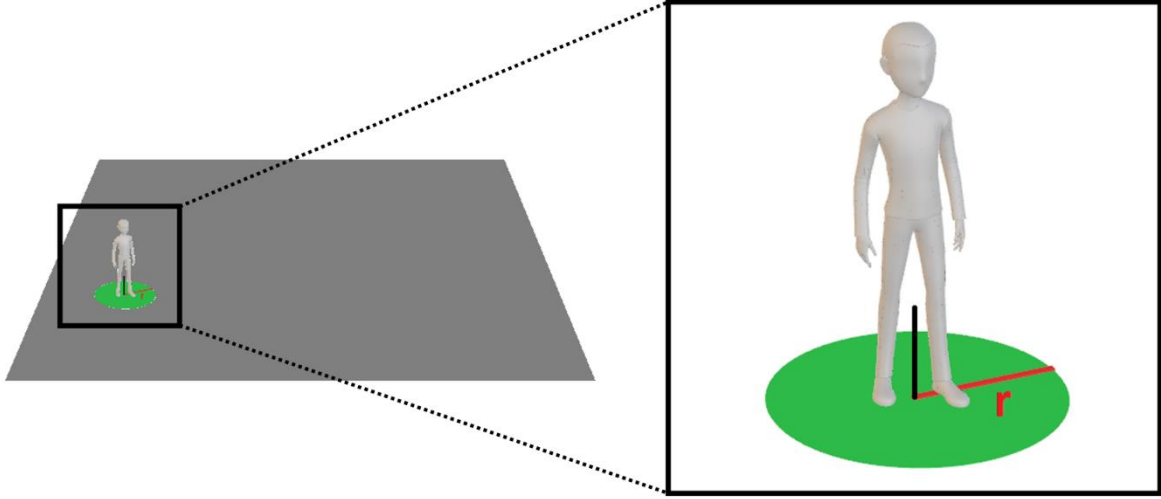


Şekil 3.2 Bireylerin benzetim alanı üzerinde kapladıkları dairesel alanın temas edeceği tüm hücrelere kimlik numaralarının yanı sıra benzetimin başlangıcına göreli ilk temas anı ve son temas anının kaydedilmesi

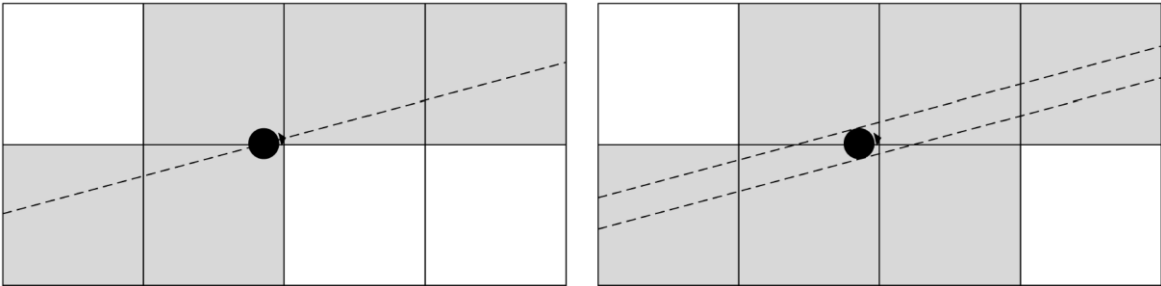
Böyle kullanışlı bir bilginin yanı sıra yeni bireyin başlangıç noktasından bitiş noktasına kadar yapacağı seyahatte ziyaret edeceği ızgara hücrelerinin ve bu hücrelere giriş-çıkış zamanlarının da bilinmesi gerekmektedir. Işın izleme için bir “Voksel Geçiş Algoritması” (*Voxel Traversal Algorithm*) [118] kullanılarak yeni bireyin yörüngesinin kesişeceği ızgara hücrelerinin belirlenmesi mümkündür. Buna ek olarak, bireylerin bu hücrelere giriş-çıkış zamanları da benzetimin başlangıcına göreli bir biçimde belirlenebilir.

Ancak ziyaret edilecek ızgara hücrelerini bireylerin doğrusal yörüngelerinin karşılık geldiği doğru parçalarını kullanarak belirlemek eksik sonuçlar üretebilir. Çünkü benzetimi oluşturan bireyler sadece noktasal birer pozisyondan ibaret olmayıp 2 boyutlu benzetim alanı üzerinde Şekil 3.3’te örneklendiği gibi merkezleri ait oldukları bireyin güncel pozisyonu ile çakışık olan r yarıçaplı diskler ile temsil edilmektedir. Bu nedenle bireyler yalnızca pozisyonlarının içinde bulunduğu ızgara hücrelerini değil kendisini temsil eden diskin temas ettiği tüm ızgara hücrelerini işgal edecektir. Diğer bir deyişle, bireyler hareketleri süresince yalnızca yörüngelerine karşılık gelen ince bir doğru parçasını değil yarıçapları ile orantılı bir alanı işgal edecektir. Eksik sonuç üretme problemini aşmak üzere getirilen basit çözüm ise ziyaret edilecek ızgara hücrelerinin bireyin yörüngesine karşılık gelen doğru parçası ile gerçekleşecek kesişimler aracılığıyla belirlenmesi yerine Şekil 3.4’te gösterildiği üzere bireyin yörüngesine paralel ve temsil edildiği diske teğet iki farklı doğru parçası için hesaplanacak kesişimler ile belirlenmesidir. Kullanılacak bu iki doğru parçasının aynı ızgara hücreleri ile kesişimleri sonucu ziyaret edilecek hücreler arasında tekrarlı sonuçlar bulunmasını önlemek adına bu iki doğru parçası tarafından üretilen çözüm kümeleri için birleşim işlemi uygulanarak sadece farklı hücreleri kapsayacak bir ortak çözüm kümesi

oluşturulur. Bu işlem esnasında aynı ızgara hücreleri için hesaplanmış olan giriş-çıkış zamanları (ziyaret zaman aralıkları) birleştirilerek her iki çözümü de kapsayacak biçimde daha geniş bir ziyaret zaman aralığı oluşturmak adına giriş zamanlarının daha düşük olanı alınırken çıkış zamanlarının daha yüksek olanı alınır.



Şekil 3.3 Bireylerin 2 boyutlu benzetim alanı üzerinde r yarıçaplı diskler ile ifade edilmesi



Şekil 3.4 Yeni birey için oluşturulan aday yolun, 2 boyutlu benzetim alanı üzerinde temas edeceği hücrelerin belirlenmesi için (Sol) eksik çözümler üretebilecek bireyin yörüngesine karşılık gelen doğru parçası yerine (Sağ) bireyin yörüngesine paralel ve temsil edildiği diske teğet iki farklı doğru parçası ile kesişimlerin hesaplanması

Yeni bireyin temas edeceği hücrelerin tam olarak belirlenmesinden sonra yalnızca bu hücreleri ziyaret edecek diğer tüm bireyler ile çarpışma testleri gerçekleştirilmesi sağlanarak gereksiz çarpışma testlerinin önemli ölçüde azaltılması olanaklıdır. Ancak bu yaklaşım dahi halen hiçbir zaman çarpışmanın mümkün olmayacağı bireylerle gereksiz çarpışma testleri gerçekleştirilmesine neden olacaktır. Bunun yerine yeni bireyin temas edeceği hücreleri

ziyaret edecek diğer bireyler ile “yaklaşık çarpışma testleri” gerçekleştirilir. Yeni bireyin ziyaret edeceği bir hücre için belirlenen ziyaret zaman aralığı ile aynı hücreyi ziyaret edecek bir başka bireyin ziyaret zaman aralığının en küçük bir bölümünün çakışması durumunda bile bu iki bireyin de belirli bir zaman dilimi boyunca aynı hücre içerisinde alan işgal edeceği söylenebilir. Bu nedenle yeni birey ile diğer bireyin yörüngeleri için, bu iki birey arasında tam manasıyla bir çarpışmaya neden olup olmayacağını belirlemek adına “kesin çarpışma testi” gerçekleştirilmesi gerekir. Ziyaret zaman aralığı yeni bireyin ziyaret zaman aralığı ile çakışmayan bireyler için ise detaylı çarpışma testleri yapılmasına gerek yoktur. Çünkü bu iki birey aynı anda aynı hücrede bulunmayacağı için ilgili hücre içerisinde çarpışmaları da mümkün değildir. Böylelikle gerçekleştirilecek çarpışma testlerinin ve dolayısıyla yeni birey oluşturma maliyetinin en düşük seviyeye düşürülmesi sağlanmaktadır.

3.1.3. Kesin Çarpışma Testlerinin Gerçekleştirilmesi

Kesin çarpışma testleri için “En Yakın Yaklaşma Noktası” (*Closest Point of Approach*) algoritması kullanılmıştır. En Yakın Yaklaşma Noktası (EYYN) algoritması normalde hareket eden iki bireyin, nesnenin veya aracın güncel pozisyonları ve değişmeyeceği kabul edilen hız vektörleri verildiğinde, bu parametrelere bağlı olarak oluşan yörüngeleri üzerindeki hareketleri boyunca pozisyonlarının birbirine en yakın olacağı zamanı hesaplamak için kullanılır. EYYN algoritması tarafından hareketin başlangıcına yani bireylerin güncel pozisyonlarında buldukları ilk ana görelilik olarak hesaplanan bu zamanı kullanarak bireylerin birbirlerine en yakın olacakları andaki pozisyonlarının hesaplanması da mümkündür. Sonrasında en yakın oldukları bu pozisyonlar arasındaki mesafe hesaplanarak benzetimi oluşturan bireylerin çapı ($2 * r$) ile karşılaştırılır. Eğer bireylerin en yakın oldukları andaki pozisyonları arasındaki uzaklık bu çap değerinden büyük veya ona eşitse bireylerin belirtilen yörüngelerinin birbirlerine göre çarpışmasız olduğu sonucuna ulaşılır. Tam aksine bireylerin en yakın oldukları andaki pozisyonları arasındaki mesafenin bu çap değerinden küçük olması durumunda ise bireyler belirtilen şekilde hareket etmeleri durumunda gelecekte bir anda mutlaka çarpışacak demektir. Ancak burada göz önünde bulundurulması gereken önemli bir nokta vardır. EYYN algoritması bireylerin belirtilen sabit hız vektörleri ile sonsuza kadar hareket edeceklerini varsaydığı için söz konusu bu çarpışma benzetim alanının sınırları dışında gerçekleşiyor da olabilir. Benzetim alanının kenarlarına ulaşan bireyler benzetimden çıkarılacağı için pratikte benzetim süresince bu bireyler arasında herhangi bir çarpışma gerçekleşmeyecektir. Bu nedenle benzetim

kapsamında ilgili bireylerin belirtilen yörüngelerinin birbirlerine göre çarpışmasız olduğunu kabul etmek yanlış olmayacaktır.

Gerçekleştirilen tüm bu kontroller sonrasında, yeni birey için belirlenen aday başlangıç pozisyonu ve hız vektörü parametreleri ile benzetimde var olan bireylerden hiçbiri arasında bir çarpışma olmaması durumunda, bu parametre değerleri ile yeni bir birey oluşturularak benzetime dâhil edilir. Yeni oluşturulan bu bireyin yörüngesi için hesaplanmış ve tüm bu kontroller sırasında kullanılmış olan ızgara hücresi ziyaret bilgileri de yeni oluşturulan bireyin kimlik numarası ile birlikte konum-zamansal dizinleme veri yapısına kaydedilir. Bu sayede konum-zamansal dizinleme veri yapısı sonraki birey oluşturma denemelerinde gerçekleştirilecek benzer hesaplamalar için yeni bireyi de içerecek biçimde güncel hale getirilmiş olur. Diğer taraftan, yeni birey ile benzetimde yer alan bireylerden herhangi biri arasında herhangi bir çarpışma tespit edilmesi durumunda ise süregelen birey oluşturma denemesi yeni birey benzetime dâhil edilmeden sonlandırılır.

3.2. Bireylerin Benzetim Alanı Üzerinde Hareket Ettirilmesi

Geliştirilen bu yönlendirmesiz kalabalık gezinim yöntemi, kalabalığı oluşturan bireylerin benzetim boyunca herhangi bir biçimde çarpışma tespiti ve çarpışmadan kaçınma manevrası yapmaması esasına dayanmaktadır. Benzetimde yer alan tüm bireylere benzetime girişleri esnasında çarpışmasız olduğu garanti edilen sabit hız vektörleri tahsis edilmektedir. Buna göre, tüm bireyler kendilerine atanan doğrusal yollar üzerinde belirlenen sabit hızlar ile benzetim boyunca yürüdüklerinde diğer hiçbir bireyle çarpışmayacakları kesindir. Bireyler herhangi bir doğrultu veya hız değişikliği yapmadığı sürece herhangi bir çarpışma gerçekleşmeyeceğine göre benzetim boyunca hiçbir birey için çarpışma tespiti ve çarpışmadan kaçınma manevraları yapılmasına da gerek yoktur.

Benzetim boyunca geçen her bir zaman adımında benzetimi oluşturan her bir bireyin pozisyonunun güncellenmesi, bireylerin çarpışmasız bir biçimde hareket ettirilmesi için yeterlidir. Bu amaçla her bir bireyin yeni pozisyonu (p_{new}), o anki pozisyonu ($p_{current}$) ile bireyin sabit hız vektörüne (v) bağlı olarak son zaman adımı içerisinde geçen süreden (Δt) kaynaklı yer değiştirmesi ile Eşitlik (3.1)'de gösterildiği gibi vektörel olarak toplanması yoluyla basitçe hesaplanabilmektedir. Böylelikle karmaşık ve maliyetli yönlendirme hesaplamaları gerektiren herhangi bir gezinim yöntemine ihtiyaç olmadan bireylerin benzetim alanı içerisinde hareketi mümkün olan en az maliyetle sağlanmış olur.

$$\overrightarrow{p_{new}} = \overrightarrow{p_{current}} + \vec{v} * \Delta t \quad (3.1)$$

Diğer taraftan, bireylerin benzetim alanı içerisindeki hareketine paralel olarak birey oluşturma sürecinde kullanılan konum-zamansal dizinleme veri yapısının da güncellenmesi gerekmektedir. Hatırlanacağı üzere bu dizinleme yapısındaki bilgiler, bireylerin başlangıç pozisyonlarından bitiş pozisyonlarına ulaşmaya kadar ziyaret edecekleri hücrelere giriş-çıkış zamanlarını saklayarak yeni oluşturulacak bireyin, aday yörüngesine göre yalnızca aynı zaman aralığında aynı hücre içerisinde bulunacağı bireyler ile çarpışma testlerine sokulmasında kullanılmaktadır. Ancak öngörülebileceği üzere benzetimde yer alan disk biçiminde bir birey, benzetim alanı üzerindeki hareketi süresince bu alanı kaplayan hayali ızgaranın hücrelerinin birinden tamamıyla çıktığında, yeni oluşturulacak bireylerden hiçbiri ile bu ızgara hücresi içerisinde çarpışması mümkün değildir. Çünkü ilgili birey artık bu hücreyi terk etmiştir ve doğrusal hareketi boyunca benzetim alanı üzerindeki görece konumu sabit olan bu hücreye bir daha uğraması söz konusu değildir. Dolayısıyla bu andan sonra oluşturulacak yeni bireylerin hiçbiri, gelecekte hiçbir zaman, sözü edilen hücre içerisinde ilgili birey ile birlikte yer alamayacağından yeni oluşturulacak bireyler ile ilgili birey arasında kesin çarpışma testleri yapılmasına da hiçbir zaman gerek olmayacaktır. Bu nedenle herhangi bir zaman adımındaki pozisyon güncellemeleri sonrasında, ziyaret ettiği bir hücreden tamamen çıkan bireylerin ayrıldığı hücre içerisinde saklanan ziyaret bilgileri silinerek konum-zamansal dizinleme veri yapısının da güncellenmesi sağlanır. Bu sayede ziyaret ettiği bir hücreden tamamen çıkan bireyler ile o andan sonra aynı hücreyi ziyaret edecek biçimde yeni oluşturulacak bireyler arasında gereksiz yere kesin çarpışma testleri yapılmasının önüne geçilmiştir. Bunun yanı sıra, geçerliliğini yitirmiş ziyaret bilgilerinin konum-zamansal veri yapısının hücrelerinde saklanmaya devam edilerek gereksiz bellek işgal etmesi ve bu veri yapısının bellek kullanımını boş yere giderek artırması da engellemiştir.

Hedefledikleri bitiş noktalarına ulaşan bireylerin ise benzetimden çıkarılması gerekmektedir. Bunu sağlamak adına bireylerin pozisyonları her güncellendiğinde, her bir bireyin güncel pozisyonundan başlayıp hedeflenen bitiş pozisyonuna (p_{goal}) uzanan doğrultu ile sabit hız vektörü arasındaki açı Eşitlik (3.2)'deki gibi hesaplanarak kontrol edilmektedir. Açının 90° 'den küçük olması durumunda hedefe uzanan doğrultu ile hız vektörünün doğrultusu birbirine paralel ve aynı yönde demektir. Bu durumda birey hedeflenen bitiş noktasına henüz ulaşamamıştır. Çünkü hareket doğrultusu bu noktaya doğru dönüktür ve benzetimin devamında bu noktaya doğru ilerlemeye devam edecektir.

$$\alpha = \arccos\left(\frac{\overrightarrow{p_{new}p_{goal}} \cdot \vec{v}}{\|\overrightarrow{p_{new}p_{goal}}\| * \|\vec{v}\|}\right) \quad (3.2)$$

Açının tam 90° olması durumunda ise bireyin hedefe uzanan doğrultu vektörü sıfırdır. Bu durumda bireyin güncel pozisyonu tam olarak bitiş pozisyonu ile çakışıktır, yani birey hedeflenen bitiş noktasına ulaşmıştır. Açının 90° 'den büyük olması durumunda ise bireyin hedefe uzanan doğrultusu ile hız vektörünün doğrultusu birbirine paralel fakat birbirine tam zıt yönlere sahiptir. Bu durumda birey hedeflenen bitiş noktasına çoktan ulaşmış ve onu geçerek benzetim alanı dışına çıkmıştır. Çünkü bu nokta hareket doğrultusuna göre bireyin tam arkasında kalmıştır. Buna göre, pozisyon güncellemeleri sonrasında bireyin hedef noktasına ulaştığı ya da onu geçtiği ($\alpha \geq 90^\circ$) ilk zaman adımında birey benzetimden çıkarılarak benzetim alanı dışında daha ileriye gitmesi engellenmiş olur.

4. GRAFİK İŞLEM BİRİMİ TABANLI TEK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ

Tez çalışması dâhilinde geliştirilen ikinci gezinim yöntemi, önceki bölümde detayları verilen AİB tabanlı TYPÇY yönteminde ortaya konan yönlendirmesiz gezinim konseptine benzer olmakla birlikte, birey oluşturma sürecinde faydalanılan GİB tabanlı çarpışmasız, doğrusal ve sabit hızlı yörünge parçası belirleme yeteneği sayesinde önemli ölçüde farklılık arz etmektedir. Geliştirilen bu gezinim yöntemine olan ihtiyacı ve bu yöntem sayesinde getirilen yenilikleri daha iyi kavramak adına önce TYPÇY bulma problemine yakından bakmak daha sonra bu probleme getirilen çözüm yaklaşımını ve bu çözüme göre geliştirilen yöntemin detaylarını incelemek yerinde olacaktır.

4.1. TYPÇY Bulma Problemi

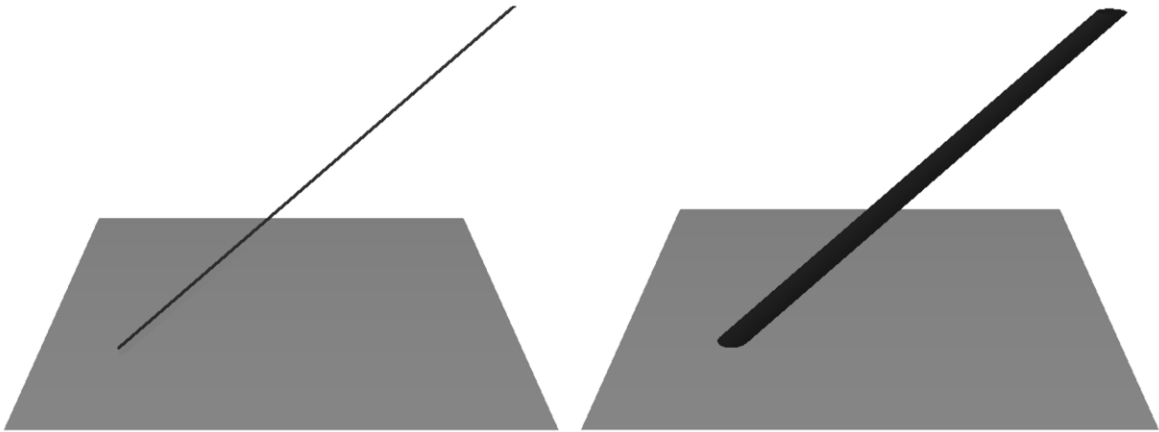
AİB tabanlı TYPÇY yöntemini konu alan önceki bölümde bahsedildiği üzere TYPÇY'lerin tespit edilmesi amacıyla rastgele başlangıç ve bitiş noktaları ile yürüme hızları seçerek, elde edilen bu hız vektörlerinin benzetimde var olan herhangi bir birey ile çarpışıp çarpışmayacağını geometrik olarak test etmek mümkündür. Ancak bu yöntem, içerdiği rastgele seçimler nedeniyle farkında olmayarak var olan bireyler ile çarpışmaya neden olabilecek doğrusal ve sabit hızlı yolları da seçebilir. Özellikle birey yoğunluğu belirli bir noktayı aştıktan sonra, yöntemin test etmek üzere herhangi bir öngörüsü olmaksızın rastgele belirleyeceği çoğu yol, çarpışmasız bir yol olmaktan ziyade benzetimde yer alan bireyler ile çarpışmaya neden olacaktır. Sistemin birey oluşturmaya çalışırken yerinde saymasına neden olan bu durum gereksiz işlem ve zaman maliyetine neden olmaktadır. Tez kapsamında çözülmeye çalışılan bir diğer problem çarpışmasız yol bulma arayışındaki rastgelelik ve bunun neden olduğu maliyetin mümkün olduğunca azaltılmasıdır.

Problemin daha iyi anlaşılmasını sağlamak üzere S olarak isimlendirilen 3 boyutlu bir konum-zaman uzayımız olduğunu varsayalım. Söz konusu bu S konum-zaman uzayında U ve V boyutları 2 boyutlu benzetim alanına karşılık gelen sıradan 2 boyutlu düzlemi ifade etmek için kullanılıyor olsun. Üçüncü boyut olan T boyutu ise diğer iki boyutun oluşturduğu düzleme dik olan zaman boyutu olsun. Benzetimde yer alan bireylerin 2 boyutlu benzetim düzlemi üzerinde bulunan başlangıç (p_{init}) ve bitiş (p_{goal}) pozisyonları Eşitlik (4.1)'de verildiği gibi ifade edilebilir. 3 boyutlu konum-zaman uzayı içerisinde yer alacak başlangıç (p_i) ve bitiş (p_g) pozisyonlarının ise Eşitlik (4.2)'de verildiği gibi tanımlanarak bireylerin doğrusal sabit hızlı yörüngelerinin S içerisinde ifade edilmesi mümkündür.

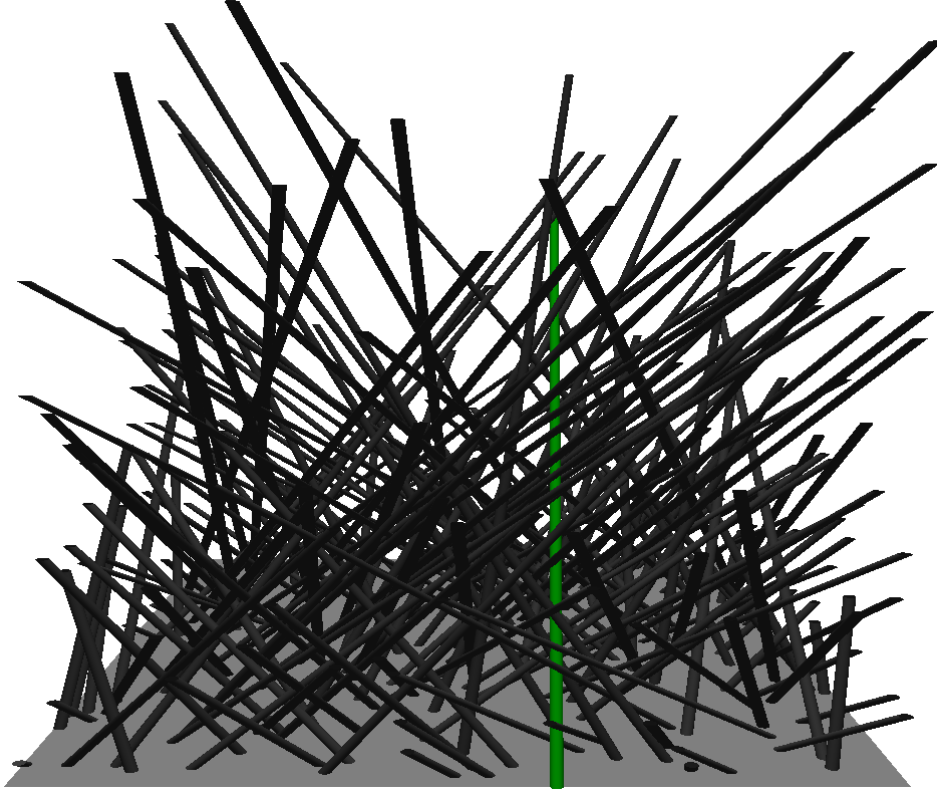
$$p_{init|goal} = \{(u, v) \in S | u \in U, v \in V\} \quad (4.1)$$

$$p_{i|g} = \{(u, v, t) \in S | u \in U, v \in V, t \in T\} \quad (4.2)$$

Bu yörüngelerin ilk bakışta S içerisinde Şekil 4.1 (Sol)'da gösterildiği biçimde doğrular oluşturacağı düşünülebilir. Ancak benzetimi oluşturan bireyler sadece noktasal birer pozisyondan ibaret olmayıp AİB tabanlı TYPÇY yöntemine benzer biçimde 2 boyutlu benzetim alanı üzerinde r yarıçaplı diskler ile temsil edilmektedir. Dolayısıyla bu bireylerin S içerisindeki konum-zamansal yörüngeleri, Şekil 4.1 (Sağ)'da verildiği gibi merkezleri az önce belirtilen doğrular ile çakışık olan doğrusal boru biçiminde hacimlere karşılık gelecektir. Benzetimde yer alan tüm bireylerin sahip oldukları yörüngelerin S içerisinde kapladıkları konum-zamansal hacimler, benzetimin herhangi bir anında bu tip borulardan oluşan kalabalık bir küme olarak düşünülebilir. Bu noktadan sonra benzetime dâhil olacak yeni bir birey oluşturma problemi, o anda S uzayında var olan yoğun boru kümesi içerisinde uygun bir boşluk bularak diğer hiçbir boruyla kesişmeyecek biçimde yeni bireyin konum-zamansal yörüngesine karşılık gelmek üzere yeni bir boru yerleştirmek olarak ifade edilebilir (Şekil 4.2). Benzetime yeni dâhil olacak bireyler için sadece benzetim alanının kenarları üzerinde oluşturularak benzetime girme gibi belirleyici bir kısıt bulunmasına karşın söz konusu 3 boyutlu boru kümesi içerisinde uygun bir boşluk bulma işi, gerçek zamanlı benzetimlerin karşılayabileceğinden çok daha yüksek bir işlem maliyetine sahiptir.



Şekil 4.1 (Sol) Bireylerin pozisyonlarının hız vektörlerine bağlı olarak konum-zamansal uzay içerisinde izleyeceği yörünge, (Sağ) Bireylerin 2 boyutlu benzetim alanı üzerinde disk biçiminde ifade edilmesinden dolayı konum-zamansal uzay içerisinde kaplayacağı boru biçimindeki hacim



Şekil 4.2 Benzetime dâhil olacak yeni bir birey oluşturmak için o anda S uzayında var olan yoğun boru (koyu gri) kümesi içerisinde uygun bir boşluk bulunarak diğer hiçbir boruyla kesişmeyecek biçimde yeni bireyin konum-zamansal yörüngesine karşılık gelmek üzere yeni bir boru (yeşil) yerleştirilmesi

4.2. Probleme Getirilen Çözüm

Sözü edilen doğrusal borulardan oluşan S uzayının sıradan bir zekâ küpü büyüklüğünde olduğunu ve bu küpü ellerimiz arasında tutabildiğimizi varsayalım. Küpün tabanı da 2 boyutlu benzetim alanına karşılılık geliyor olsun. Bu küpü ellerimiz arasında serbestçe döndürerek taban kenarları üzerindeki herhangi bir bakış noktasından içerisine doğru bakma imkânımız da olsun. Böyle bir senaryoda baktığımız noktadan tıpkı kapıdaki bir anahtar deliği misali küpün diğer tarafını görebildiğimiz durumlarda bakış noktasından başlayıp küpün diğer tarafına açılan doğrusal bir oyuk var demektir. Bu oyuklardan yeterli genişlikte olanlarının içerisine yeni bir doğrusal boru yerleştirmek de mümkün olacaktır. Yeni yerleştirilen bu borunun yeni oluşturulacak bireyin konum-zamansal yörüngesine karşılık geldiği düşünüldüğünde, benzetim alanının bir kenarından girip diğerinden çıkan ve sabit hızla hareket ederken var olan hiçbir birey ile çarpışmayan yeni bir bireyin benzetime dâhil edilmesi kolaylıkla mümkün olacaktır.

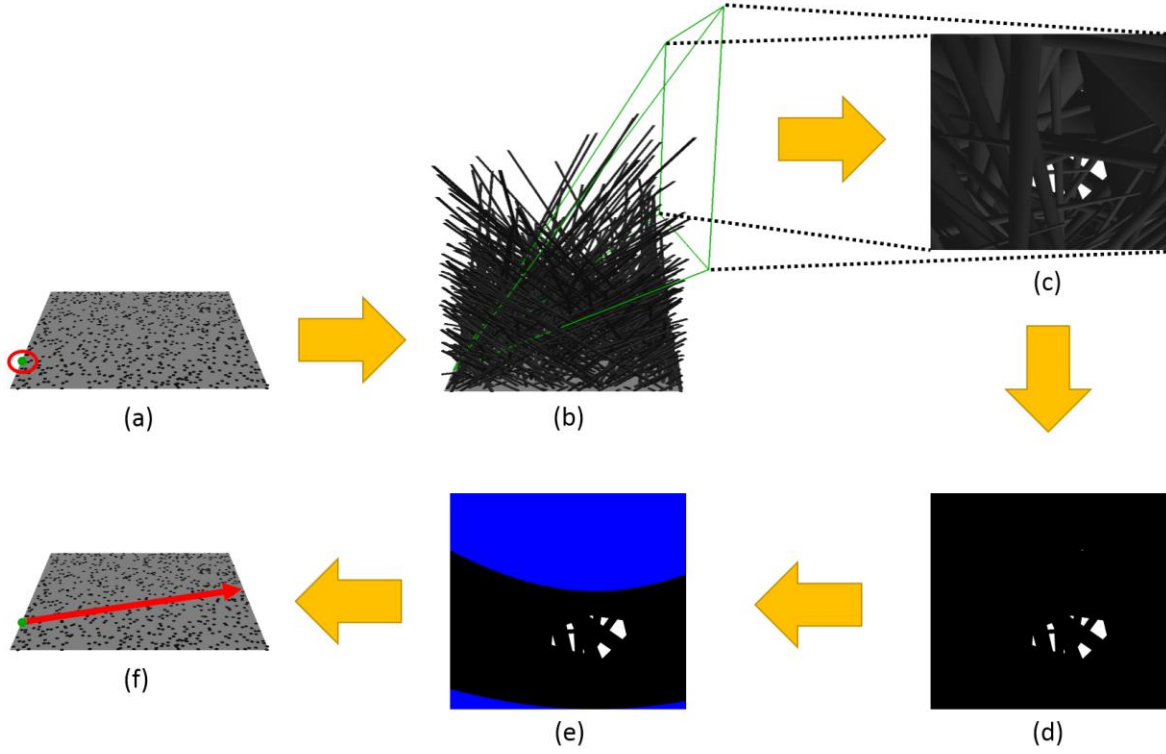
Anlatılan bu hayali senaryodan esinlenilerek GİB'nin hızlı görselleştirme yeteneklerinin yardımıyla bu gibi bir kurguyu gerçekleştirmenin mümkün olabileceği düşünülmüştür. GİB ile bireylerin yörüngelerini temsil eden doğrusal borulardan oluşan ve sözü edilen hayali S uzayına karşılık gelen 3 boyutlu sanal bir sahne yapılandırmak oldukça kolaydır. Bu sahne içerisinde kamera pozisyonunu ve perspektif özelliklerini ayarlayarak ihtiyaç duyulduğu gibi bir bakış elde etmek de mümkündür. Sonrasında sahenin bu bakış açısında görselleştirilmesiyle meydana gelen görüntü bir resim olarak kaydedilebilir. Resmi oluşturan pikselleri tarayarak resim üzerinde yer alan ve olası çözümlere karşılık gelen boşluklar tespit edilebilir. Bu boşluklar kameranın bakış noktasından başlayıp S uzayının diğer tarafına kadar uzanan ve yeni bir doğrusal boru yerleştirilebilecek bir oyuk olduğu anlamına gelmektedir. Yeterli genişlikte bir boşluğu kullanarak kameranın pozisyonuna karşılık gelen bir başlangıç noktası, oyuğun diğer tarafında boşluğun merkezine karşılık gelen bir bitiş noktası ve bu iki nokta arasındaki doğrunun benzetim zeminine göre eğiminden faydalanarak bir hız değeri elde etmek mümkündür. Bu sayede benzetime katılacak yeni bir birey için gerekli tüm parametreler belirlenmiş olur.

Geliştirilen bu yenilikçi birey oluşturma yaklaşımı ile TYPÇY bulma probleminin rastgelelikten olabildiğince uzak ve daha verimli bir biçimde çözülmesi sağlanmıştır. Tez kapsamında geliştirilen GİB tabanlı TYPÇY yönteminin temel olarak iki ana bileşenden oluştuğu söylenebilir. Bunlardan ilki yukarıda sözü edilen GİB tabanlı yeni birey oluşturma yaklaşımını kullanarak benzetime katılacak bireyler için başlangıç pozisyonu, bitiş pozisyonu ve sabit yürüme hızı değerlerini belirlemektedir. İkincisi ise benzetim boyunca geçen her bir zaman adımı içerisinde benzetimi oluşturan bireylerin pozisyonlarının yürüme hızlarına bağlı olarak güncellenmesinden sorumludur.

4.3. Yeni Birey Oluşturma Sürecinin Detayları

Benzetime katılmak üzere yeni bir birey oluşturmak için bu bireye çarpışmasız sabit bir hız vektörü atanması gerekmektedir. Bu amaçla ilk önce benzetim alanının rastgele seçilecek bir kenarı üzerinde yer alan yine rastgele bir başlangıç pozisyonu seçilir. Aslına bakılırsa tez kapsamında geliştirilen GİB tabanlı yeni birey oluşturma yönteminin rastgele olan tek kısmı başlangıç pozisyonunun belirlenmesidir. Daha sonra GİB'nin “görselleştirme iş hattından” (*rendering pipeline*) faydalanılarak sabit bir hız vektörü belirlenir. Belirlenen sabit hız vektörünün bir sonucu olarak oluşan doğrusal yolun benzetim alanının diğer bir kenarı ile kesiştiği noktada bir bitiş pozisyonu meydana gelecektir.

Yeni oluşturulacak bireyler için çarpışmasız sabit hız vektörü belirleme süreci birbirini izleyen iki alt süreçten oluşmaktadır. Bunlardan ilki benzetimde var olan bireylerin konum-zamansal görselleştirilmesi, ikincisi ise görselleştirme sonrası olası çözümlerin analiz edilmesidir. Çarpışmasız hız vektörü belirleme süreci Şekil 4.3'te resimli olarak gösterilmiştir.

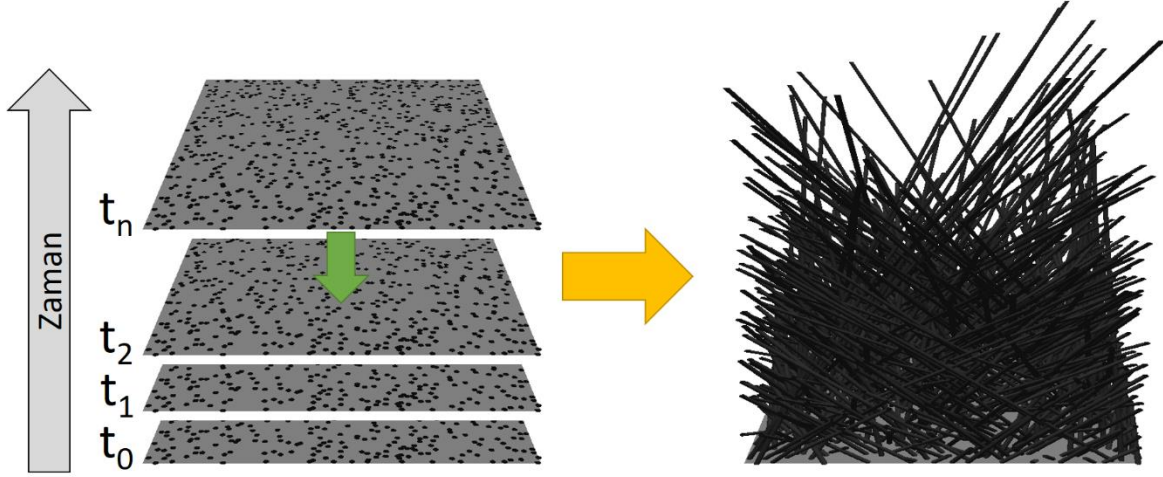


Şekil 4.3 Çarpışmasız hız vektörü belirleme süreci (a) Benzetim alanının kenarlarından biri üzerinde rastgele bir başlangıç pozisyonu (yeşil nokta) seçilmesi (b) Kameranın ayarlanması ve var olan bireylerin konum-zamansal borularının görselleştirilmesi (c) Oluşturulan konum-zamansal sahnenin kameranın bakış açısından görünümü (d) Sahnenin kameranın bakış açısından elde edilen siyah beyaz resmi (e) Ön tanımlı hız aralığı içerisinde sonuçlar üreten çözümlerin (mavi olmayan pikseller) belirlenmesi (f) Beyaz bir piksel seçerek çarpışmasız sabit hız vektörünün (bitiş pozisyonu ve yürüme hızı) belirlenmesi

4.3.1. Bireylerin Konum-Zamansal Görselleştirilmesi

Daha önce de belirtildiği üzere bireyler 2 boyutlu benzetim alanı üzerinde yüksekliği olmayan diskler ile ifade edilmektedir. Benzetimde var olan tüm bireylerin konum-zamansal olarak görselleştirilmesi, her bir zaman adımında benzetim alanının tamamının kuş bakışı ortografik iz düşüm resminin alınması ve bu resimlerin zaman adımı sırasına göre üst üste

yığılması olarak hayal edilebilir. Bireyleri içi dolu daireler olarak gösteren 2 boyutlu bu resimler, en sonunda bireylerin hız vektörlerine göre o anki konumlarından bitiş pozisyonlarına doğru eğimli doğrusal silindirik borulardan oluşan 3 boyutlu bir sahne oluşturacaktır. Bu senaryonun bir örneği Şekil 4.4'te resmedilmiştir.



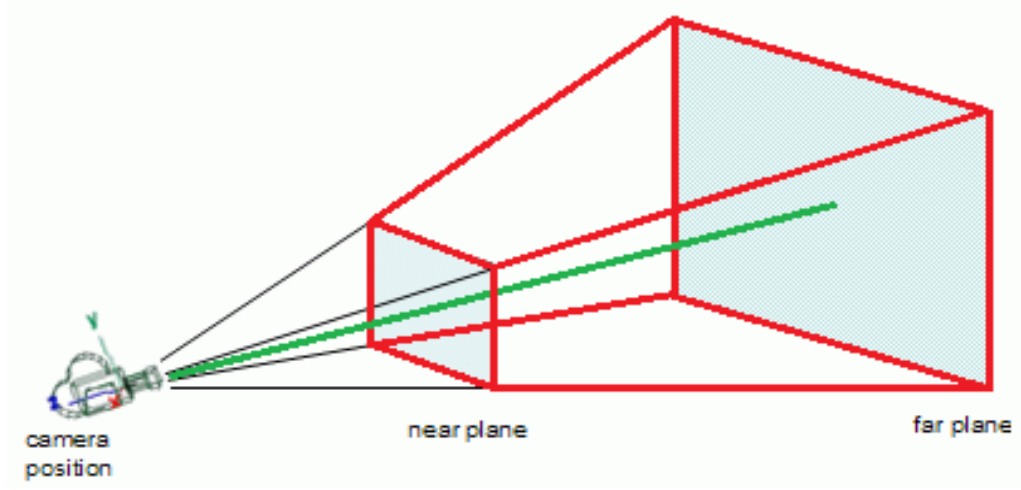
Şekil 4.4 Benzetim alanının kuşbakışı ortografik iz düşüm resimlerinin üst üste yığılmasıyla oluşan silindirik borulardan meydana gelen 3 boyutlu sahne

Bu işlem kulağa çok zaman ve maliyet gerektiriyor gibi gelebilir ancak bireylerin anlık pozisyonları, bitiş pozisyonları ve hızları bilindiği sürece her bireyi 3 boyutlu silindirik bir model kullanarak görselleştirmek ve böyle bir sahneyi oluşturmak oldukça kolaydır. Bu amaçla taban çapı ve boyu 1 birim olan 3 boyutlu bir silindir modeli kullanılmıştır. Tüm bireyler bu silindir modelinin belirli oranlarda ölçeklendirilip eğilmesiyle görselleştirilmiştir. X , Y ve Z boyutlarından oluşan 3 boyutlu görselleştirme uzayının koordinat sistemi ile S uzayının koordinat sisteminin orijin noktalarında çakışık olduğu ve aynı zamanda karşılıklı olarak X ile U , Z ile V ve Y ile T eksenlerinin de çakışık olduğu varsayılmıştır. Bu durumda 2 boyutlu benzetim alanı ile görselleştirme uzayının X - Z düzlemi de çakışıyor olacaktır. Ayrıca görselleştirme uzayındaki 1 birimlik uzunluğun 1 metreye karşılık geldiği kabul edilmiştir. Bireylerin görselleştirilmesi işleminin algoritması aşağıda verilmiştir.

- Öncelikle sahne arka plan rengi beyaz olarak ayarlanır.
- Sahneye yerleştirilecek silindirlerin rengi ise siyah olarak ayarlanır.
- Benzetimde yer alan her bir birey için:

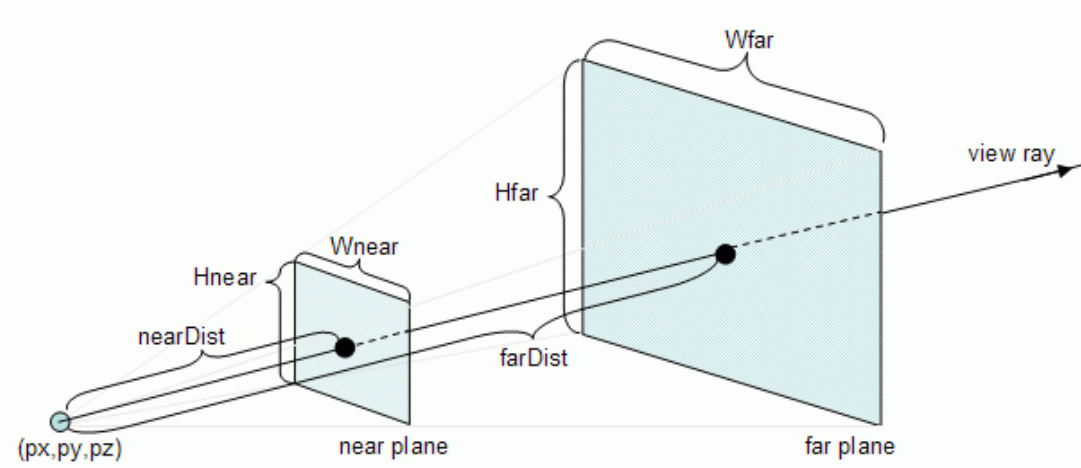
- Bireyin güncel pozisyonu ile bitiş pozisyonu arasındaki mesafe hesaplanır.
- Bu mesafe bireyin yürüme hızına bölünerek bireyin hedeflediği bitiş noktasına ne kadar sürede ulaşacağı bulunur.
- Silindir modelinin boyu bu süreye eşit olacak şekilde silindir Y boyutunda ölçeklenir.
- Silindir X ve Z boyutlarında yarıçapı $2 * r + \varepsilon$ olacak şekilde ölçeklenir.
- Silindirin taban dairesinin merkezi bireyin güncel pozisyonuyla çakışık olacak biçimde silindir benzetim alanına tabanı üzerine yerleştirilir.
- Silindir modelinin tabanı sabit kalmak suretiyle tavan dairesinin merkez noktasının iz düşümü bireyin bitiş pozisyonu ile çakışık olacak şekilde silindir eğilir.

İhtiyaç duyulan 3 boyutlu konum-zamansal sahnenin yaratılmasından sonra yeni oluşturulmaya aday birey için rastgele seçilen başlangıç pozisyonundan bu sahneyi görececek bir kamera ayarlaması yapılması gerekir. Bu bakış açısı ile sahnenin iz düşüm resminin kaydedilmesi işlemi olası hız vektörü çözümlerinin bulunması için kritik bir süreç olup bu amaçla kameranın pozisyonunun ve perspektif özelliklerinin doğru ayarlanması şarttır. Kamera ayarlamalarının anlatımına geçmeden önce kameranın çalışma prensiplerini kısaca özetlemek yerinde olacaktır. 3 boyutlu uzayda kameranın görebileceği sınırlı hacim “görüş hacmi” (*viewing frustum*) olarak anılır. *Frustum* kelime anlamı olarak koni ya da piramit gibi şekillerin tabanına paralel bir düzlem ile tepeden kesilmesi sonucu oluşan geometrik şekillere verilen addır. Perspektif kameranın kullandığı görüş hacmi, tepe noktası kameranın pozisyonu ile çakışık olan, tabanının merkez noktası kameranın bakış doğrultusu üzerinde yer alan ve kameranın bakış yönü bu merkez noktasına doğru olan bir piramit biçiminde düşünülebilir (Şekil 4.5).

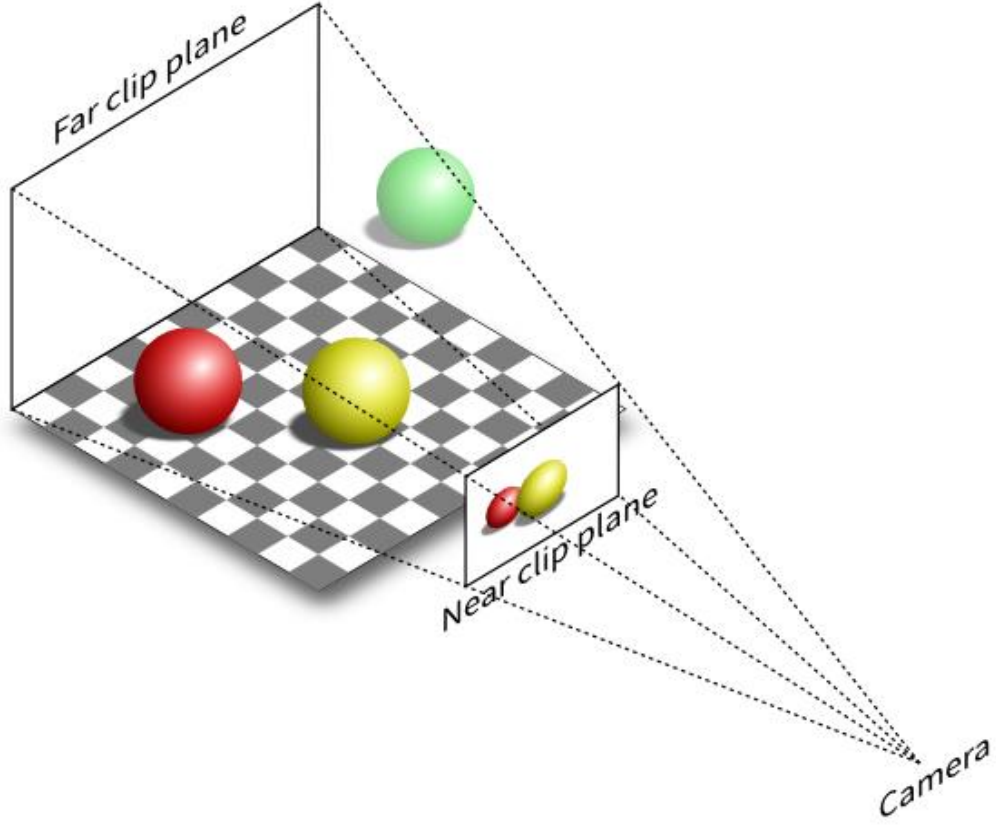


Şekil 4.5 Perspektif kamerada görüş hacmi [119]

Kameranın gördüğü hacim altı adet “görüş hacmi düzlemi” ile sınırlanmaktadır (Şekil 4.6). Görüş hacminin büyüklüğünü bu düzlemlerin boyutları ve kameraya olan uzaklıkları belirler. Bu hacim içerisinde yer alan nesnelerin kameranın bakış doğrultusuna dik ve kameraya yakın olan “yakın düzlem” (*near plane*) üzerine perspektif iz düşümü yapılarak kameranın gördüğü görüntünün oluşturulması sağlanmaktadır. Görüş hacminin dışında kalan nesneler ise yakın düzlem üzerinde iz düşüm oluşturamayacağı için kamera tarafından görülemez (Şekil 4.7). Kameranın bakış doğrultusuna dik ve kameraya uzak olan “uzak düzlem” (*far plane*) ise kameranın bakış doğrultusu üzerinde görebileceği en uzak hat olan “ufuk sınırı” gibi görev yapar.



Şekil 4.6 Perspektif kamerada görüş hacminin büyüklüğünü belirleyen faktörler [120]



Şekil 4.7 Perspektif kamerada görüntü oluşumu [121]

Kamera pozisyonu geçici olarak, üzerinde yer alacağı kenarın tam ortasında olacak şekilde yerleştirilir. Görüş hacminin yakın düzlemi, kameraya yakın alanlarda görüş kaybını önlemek üzere kameraya olabildiğince yakın (0.001 m) ayarlanır. Görüş hacminin uzak düzlemi ise geçici olarak, benzetim alanında kameranın konumlandırılacağı kenarın karşısındaki kenar üzerinde yer alacak biçimde kameradan benzetim alanının derinliği (d_{env}) kadar uzağa yerleştirilir (Şekil 4.8). Kameranın görüş hacminin sol ve sağ düzlemleri arasındaki “yatay görüş açısı” (*horizontal field of view*) uzak düzlem tam benzetim alanı genişliğinde (w_{env}) olacak şekilde ayarlanır (Şekil 4.8). Kameranın benzetim alanı üzerinde yerleştirileceği kenara bağlı olarak minimum ve maksimum seyahat mesafeleri hesaplanır. Minimum seyahat mesafesi (d_{min}) kameranın yerleştirileceği kenar ile onun tam karşısındaki kenar arasındaki uzaklığa karşılık gelir (Şekil 4.8). Benzer biçimde, maksimum seyahat mesafesi (d_{max}) ise kameranın yerleştirileceği kenar ile onun tam karşısındaki kenar arasındaki köşegenin uzunluğuna karşılık gelir (Şekil 4.8).



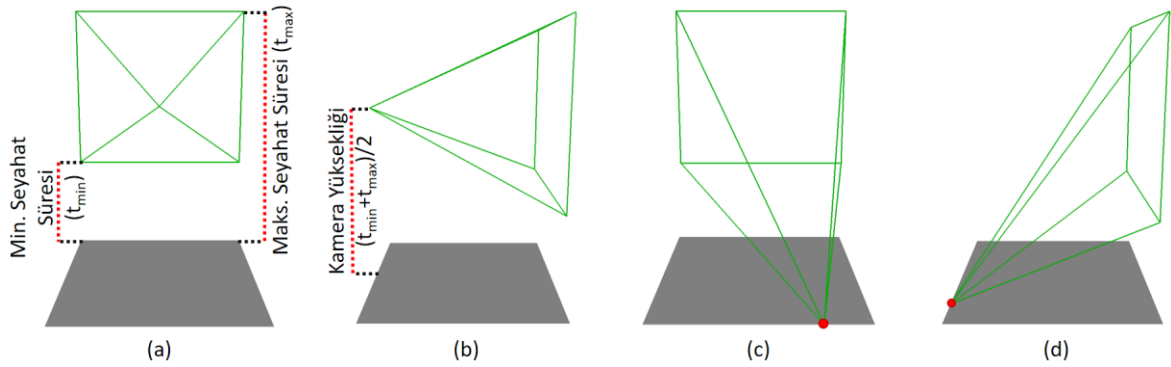
Şekil 4.8 Benzetim alanının boyutları ve kameranın görüş hacminin benzetim alanı üzerine iz düşümü

Benzetimi oluşturan bireylerin yürüme hızları ön tanımlı $[s_{min}, s_{max}]$ değer aralığı ile sınırlanmıştır. Bu sayede bireylerin benzetim alanı üzerinde başlangıç pozisyonlarından bitiş pozisyonlarına ulaşmaya kadar geçirebileceği minimum ve maksimum seyahat süreleri (t_{min} ve t_{max}) de Eşitlik (4.3) ve (4.4)'teki gibi hesaplanabilir. Kameranın görüş hacminin tüm seyahat süre aralığını $[t_{min}, t_{max}]$ kapsamaları için görüş hacminin üst ve alt düzlemleri arasındaki “dikey görüş açısı” (*vertical field of view*) uygun biçimde ayarlanarak uzak düzlemin tam $(t_{max} - t_{min})$ yüksekliğinde olması sağlanır (Şekil 4.9 (a)). Kamera pozisyonunun yüksekliği $(t_{min} + t_{max}) / 2$ olacak şekilde ayarlanarak tüm görüş alanı da aynı miktarda yukarı doğru kaydırılmış olur. Böylelikle zaman boyutuna karşılık gelen Y ekseninde uzak düzlemin alt kenarının minimum seyahat süresi kadar, üst kenarının ise maksimum seyahat süresi kadar yüksekte olması sağlanır (Şekil 4.9 (a) ve (b)). Kameranın pozisyonunun seçilen başlangıç pozisyonu ile çakışık duruma getirilmesi gerekmektedir (Şekil 4.9 (c)). Bu amaçla kameranın “iz düşüm matrisi” (*projection matrix*), görüş hacmi piramidini eğerek tepe noktasını başlangıç pozisyonu ile çakıştıracak uygun bir “eğme” (*shear*) matrisi ile çarpılır. Bu eğme matrisi yeni birey için seçilen başlangıç kenarına ve bu kenar üzerinde bulunan başlangıç pozisyonuna göre uygun eğmeyi sağlayacak biçimde oluşturulur. Görüş alanı eğildikten sonra yakın ve uzak düzlemlerin halen 2 boyutlu benzetim alanına dik olması, bulunacak olası çözümlerin yürüme hızı hesaplamalarının doğru yapılmasını sağlaması adına vurgulanması gereken önemli bir husustur (Şekil 4.9 (d)). Son olarak, kameranın görüş hacminin uzak düzlemi 10 m ileriye ötelenip nihai konumuna

yerleştirilerek bu kenar üzerinde yer alabilecek bireylerin görüş hacmi tarafından bütünüyle kapsanması garanti edilmiş olur (Şekil 4.8).

$$t_{min} = \frac{d_{min}}{s_{max}} \quad (4.3)$$

$$t_{max} = \frac{d_{max}}{s_{min}} \quad (4.4)$$



Şekil 4.9 Kameranın pozisyonunun ve görüş hacminin ayarlanması

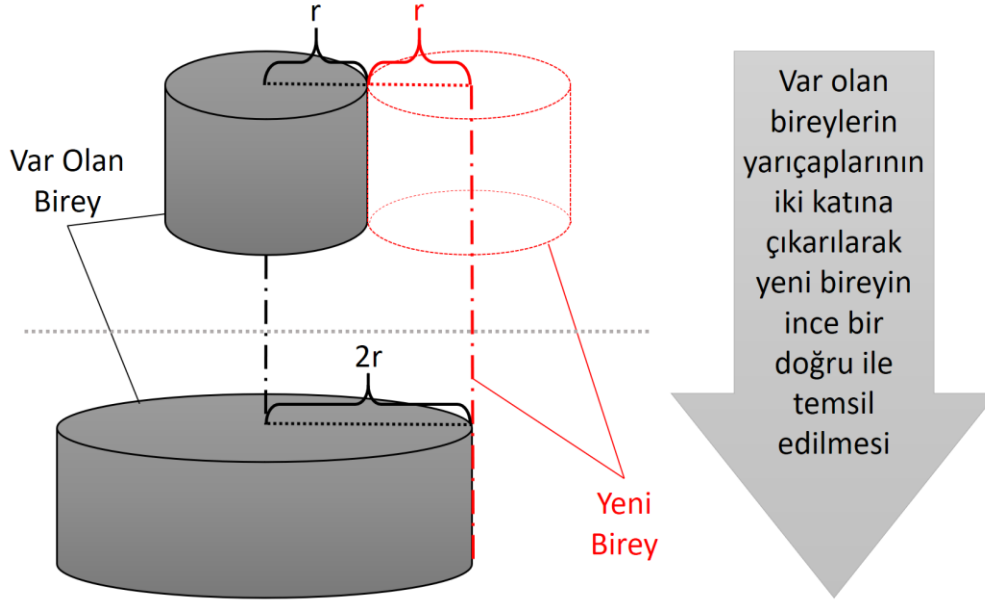
Kameranın yerleştirildiği başlangıç pozisyonundan başlayan ve benzetim alanı ile belirli bir açı yaparak karşı kenara doğru uzanıp uzak düzlemi kesen doğrular, oluşturulacak yeni birey için olası birer çözüme karşılık gelmektedir. Bu doğruların benzetim alanı üzerindeki izdüşümlerinin karşı kenarı kestiği nokta bitiş pozisyonunu, benzetim alanı ile yaptığı eğimin çarpma işlemine göre tersi ise yürüme hızını verecektir. Yeni oluşturulacak birey için belirlenecek bitiş pozisyonu karşı kenar üzerinde, sabit yürüme hızı da daha önce tanımlanan yürüme hızı aralığı içerisinde olmalıdır. Bunu sağlamak üzere olası çözüm doğrularının görüş hacmi içerisinde ve $[1/s_{max}, 1/s_{min}]$ aralığında eğimlere sahip olması gerekmektedir. Bu amaçla uygun biçimde ayarlanan görüş hacmi sayesinde belirlenen aralıkların dışında bitiş noktası ve hız değeri üreten geçersiz çözümlerin çok büyük bir bölümünün hiç hesaba katılmayacağı garanti edilmiş olur.

Benzetimde var olan bireylerin konum-zamansal borularından oluşan 3 boyutlu sahnenin “ekran dışı” (*offscreen*) görselleştirilmesi için GİB’nin belleği içerisinde görüntü verilerinin yazılacağı bir “resim karesi arabellek nesnesi” (*frame buffer object*) oluşturulması gerekmektedir. Benzetimin en başında henüz hiçbir birey oluşturma işlemi yapılmadan önce

tanımlanan bu resim karesi arabellek nesnesi (RKAN), kameranın görüş hacminin yakın düzlemi üzerine alınacak görüntü iz düşümünün saklanacağı bellek alanıdır. Kayan noktalı sayı formatında kırmızı, yeşil ve mavi (KYM) renk bileşenlerinden oluşan bu RKAN, kameranın görüş hacminin uzak düzleminin boyutlarıyla orantılı olacak biçimde $w_{env} * k$ genişliğine ve $(t_{max} - t_{min}) * k$ yüksekliğine sahip olacak şekilde oluşturulur. Burada kullanılan k değeri görselleştirme sonrası elde edilecek iz düşüm resminin, yani çözüm uzayının çözünürlüğünü ayarlamak üzere kullanılan sabit bir ölçeklendirme çarpanıdır. Değiniilmesi gereken önemli bir nokta ise, ölçeklendirme çarpanı artırılarak çözüm uzayının çözünürlüğü artırıldığında daha hassas ve isabetli sonuçlar alınacağıdır. Ancak yine de artan çözünürlüğe bağlı olarak görselleştirme iş hattı üzerinde gerçekleştirilecek işlemlerin maliyetinin de artacağı göz önünde bulundurularak k değerinin ihtiyaca uygun biçimde dikkatli seçilmesi önerilmektedir.

Bireylerin konum-zamansal görselleştirilmesinin sonunda RKAN içerisinde sahnenin kamera bakış açısından elde edilmiş siyah beyaz bir iz düşüm resmi oluşur (Şekil 4.3 (d)). Bu resimde yer alan siyah renkli pikseller var olan bireylerin konum-zamansal borularına karşılık gelmektedir. Yani bu siyah piksellere göre bir bitiş pozisyonu ve bir yürüme hızı seçerek yeni bir birey oluşturmak, yeni bireyin eninde sonunda benzetimde yer alan en az bir veya daha fazla bireyle çarpışacağı anlamına gelir. Diğer yandan, beyaz renkli pikseller doğrudan boru kümesinin diğer tarafından görünen boşluğa karşılık geldiği için beyaz piksellerden oluşan yeterli büyüklükte bir boşluğa yeni bireyi ifade eden bir konum-zaman borusu yerleştirilebilir. Kısacası beyaz pikselleri kullanarak yeni bireyler için çarpışmasız hız vektörleri belirlemek mümkündür.

Birey oluşturmak üzere resim üzerinde beyaz piksellerden oluşan yeterince geniş bir boşluk aramak yerine her bir beyaz pikselin konum-zaman uzayında olası bir çarpışmasız yola karşılık geldiğini garanti etmek ve çözüm olarak tek bir beyaz piksel kullanmak da olanaklıdır. Bu amaçla daha önce de belirtildiği gibi bireylerin konum-zamansal görselleştirilmesi esnasında KZYE'ye karşılık gelen doğrusal borularının yarıçapları $2 * r + \epsilon$ olacak şekilde ölçeklenmiştir. Bu sayede yeni oluşturulacak bireyin dairesel çapı önceden hesaba katılarak sadece konum-zamansal pozisyonuna karşılık gelen sıradan bir doğruyla ifade edilmesi sağlanmış olur (Şekil 4.10). Teorik olarak var olan bireylerin yarıçapını iki katına çıkarmak bu işlem için yeterli gibi görünse de pratikte “pikselleştirme” (*rasterization*) aşamasında meydana gelen “kayan noktalı sayı” (*floating point number*) hataları nedeniyle ayarlanabilir küçük bir değeri ifade eden ϵ değeri de ölçeklendirmeye eklenmiştir.



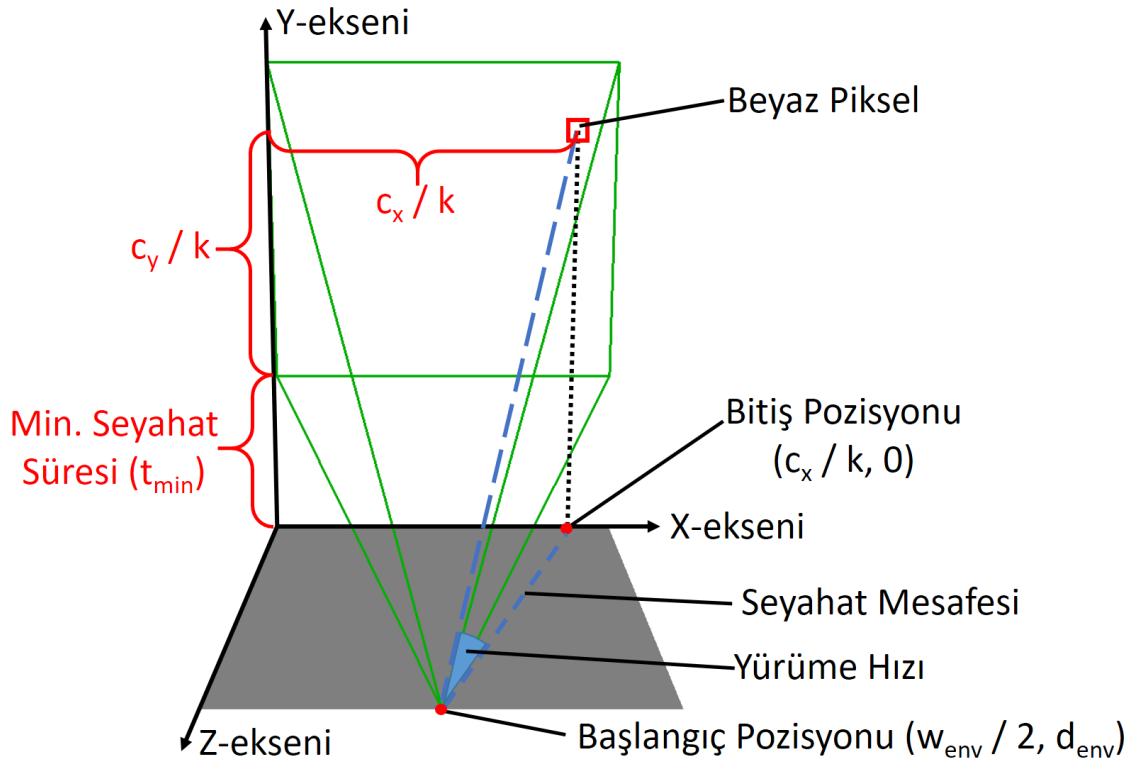
Şekil 4.10 Yeni bireyin dairesel çapının önceden hesaba katılarak sadece bir doğru ile ifade edilmesi

4.3.2. Olası Çözümlerin Analiz Edilmesi

Bireylerin konum-zamansal görselleştirilmesi sürecinin çıktısı olarak 2 boyutlu bir görüntü iz düşün resmi elde edilir. Siyah beyaz bu resimde yer alan beyaz pikseller olası çözümleri ifade eder. Resim içerisindeki beyaz piksellerin resim çerçevesinin köşesine göreli yatay (c_x) ve dikey (c_y) koordinatlarından Eşitlik (4.5)'i kullanarak 2 boyutlu benzetim alanı üzerinde bulunan bir bitiş pozisyonu ve Eşitlik (4.6)'yı kullanarak çarpışmasız bir hız vektörü değeri elde etmek mümkündür. Başlangıç pozisyonu olarak 2 boyutlu benzetim alanı üzerindeki $(w_{env}/2, d_{env})$ noktasının seçildiğini varsayan örnek bir senaryo Şekil 4.11'de gösterilmiştir. Burada dikkat edilmesi gereken önemli bir nokta ise beyaz pikseller kullanılarak elde edilecek hız vektörlerinin hepsi çarpışmasız olmasına rağmen seçilmiş olan başlangıç pozisyonuna bağlı olarak bazılarının belirlenen $[s_{min}, s_{max}]$ yürüme hızı aralığı dışında hız değerleri ($s = \|v\|$) üretebileceğidir. Bu nedenle, aralığın dışında istenmeyen hız değerleri üreten olası çözümlere karşılık gelen beyaz piksellerin (Şekil 4.3 (e)'de mavi pikseller ile gösterilen alanda kalanlar) hesaba katılmaması gerekmektedir.

$$p_{goal} = \begin{cases} \left(\frac{c_x}{k}, 0\right) & : p_{init} = ([0, w_{env}], d_{env}) \\ \left(w_{env} - \frac{c_x}{k}, d_{env}\right) & : p_{init} = ([0, w_{env}], 0) \\ \left(0, d_{env} - \frac{c_x}{k}\right) & : p_{init} = (w_{env}, [0, d_{env}]) \\ \left(w_{env}, \frac{c_x}{k}\right) & : p_{init} = (0, [0, d_{env}]) \end{cases} \quad (4.5)$$

$$v = \frac{\overrightarrow{p_{init}p_{goal}}}{\frac{c_y}{k} + t_{min}} \quad (4.6)$$



Şekil 4.11 Seçilen beyaz bir piksele göre bitiş pozisyonu ve yürüme hızı değerini belirlenmesi

Her bir beyaz piksel olası bir çözümü ifade etmesine rağmen elde edilen resmin satır satır taranarak beyaz piksellerin aranması mantıklı olmadığı gibi her birey oluşturma denemesi için gerçek zamanda gerçekleştirilmesi de mümkün değildir. Ayrıca satır satır tarama için tüm resmin GİB belleğinden AİB belleğine kopyalanması amacıyla yapılacak veri transferi ve AİB tarafından bu resmin işlenmesi de fazladan maliyet getiren işlemlerdir. Buna ek

olarak, satır satır tarama esnasında karşılaşılabilecek ilk beyaz pikselin çözüm olarak seçilmesi, benzetim alanı üzerinde oluşturulacak bireylerin tekdüze olmayan bir biçimde dağılımına neden olacaktır. Bu nedenle komşu beyaz piksel kümelerinin merkez noktalarını hesaplamak üzere GİB'yi kullanan bir yöntem önerilmiştir. Bu sayede sadece GİB belleğinden AİB belleğine transfer edilecek veri miktarını azaltmakla kalmayıp GİB'nin “tek komut çoklu veri” (*single instruction multiple data - SIMD*) temeline dayanan paralel özelliklerinden de faydalanılmıştır.

Aslında satır satır pikseller ile ifade edilen resim üzerindeki beyaz alanların tespit edilmesi problemi, bilindik “bağlantılı bileşen etiketleme” (*connected component labeling*) teknikleri kullanılarak çözülebilir [122]. Bağlantılı bileşen etiketleme (BBE) görüntü işleme ve bilgisayarlı görü alanlarında sayısal resimlerin bağlantılı bölümlerinin tespit edilmesi ve her bir bağlantılı bölümün biricik bir etiket ile etiketlenmesi amacıyla kullanılmaktadır [123]–[125]. Her bir pikselin komşuluğunu belirlemek üzere yatay ve dikey eksenindeki dört komşu pikseline bakmak mümkün olduğu gibi çaprazdaki komşularını da dâhil ederek “8 piksel komşuluğu” da kullanılabilir. BBE kullanarak sadece siyah beyaz resimlerin işlenmesi değil renkli resimlerin veya her bir eleman için daha yüksek boyutluluğa sahip verilerin ele alınması da mümkündür [126]. Wu ve arkadaşları [127] BBE metotlarını çok geçişli, çift geçişli ve tek geçişli olmak üzere üç başlık altında sınıflandırmışlardır. Çok geçişli metotlar piksellerin yerel komşuluğunu kullanarak minimum etiketlerini belirlerken resmin içeriğine bağlı olarak etiketlerin genel minimum değerlerine ulaşmak adına çok sayıda yineleme gerektirir [128]. İki geçişli metotlar resim üzerinde iki farklı geçiş yapar [129]. İlk geçişte komşu pikseller için etiket denklikleri kaydedilir ve piksellerin nihai etiketleri bu denklik zincirleri çözümlenerek belirlenir. İkinci geçişte ise nihai etiket değerleri karşılık geldikleri piksellere atanır. Kenar izleme [130] gibi tek geçişli metotlar satır satır tarama benzeri çalışarak resmi sadece bir kez işler ve bağlantılı bölümleri uygun biçimde etiketler. Diğer taraftan, GİB kullanarak yapılan genel amaçlı hesaplama alanındaki son gelişmeler GİB'nin paralel mimarisinden faydalanabilen BBE algoritmaları geliştirilmesinin de önünü açmıştır [131]–[133]. Bu algoritmalar genel olarak benzer bir paralel tarama adımı yardımıyla her bir pikselin komşuluğunu kontrol ederek piksellere geçici etiketler atar. Ancak daha sonra bu algoritmaların, bağlantılı bölgeleri birleştirmek üzere çok sayıda yineleme yapması gerekir. Çünkü tek bir birleştirme geçişinden sonra her bir bağlantılı bileşen için tek bir denklik zinciri oluşturulduğunu garanti edemezler. Ayrıca

tüm bağlantılı bölgeleri birleştirmek için gerekli olacak geçiş sayısı resmin içeriğine bağlı olmakla birlikte bu sayının, işlemin en başında belirlenmesi mümkün değildir.

Nitekim bizim bağlantılı bileşenlerin doğru ve eksiksiz etiketlenmesinden ziyade beyaz piksel gruplarının merkez noktalarının belirlenmesine ihtiyacımız vardır. Bu amaçla grafik iş hattı kontrol dili “OpenGL Shading Language” (GLSL) kullanılarak GİB’nin görselleştirme iş hattını kullanan iki “Grafik İşlem Biriminde Genel Amaçlı Hesaplama” (*General Purpose Computing on Graphics Processing Unit - GPGPU*) geçişinden oluşan paralel bir algoritma geliştirilmiştir. Söz konusu bu algoritma Hacımeroglu ve arkadaşları [73] tarafından benzer hız vektörlerine sahip olan ve birbirlerine yakın yürüyen kalabalık benzetimi bireylerinin oluşturduğu kümelerin belirlenmesi için önerilen algoritmaya benzetilmektedir.

İlk geçişte konum-zamansal sahnenin iz düşüm resmini içeren RKAN, “doku resmi” (*texture image*) olarak ayarlanır. İz düşüm resmi ile aynı boyutlara sahip diğer bir RKAN oluşturulur ve görselleştirme sonuçlarının yazılacağı “çizim belleği” (*draw buffer*) olarak atanır. Görüş alanı doku resminin tamamını kaplayacak büyüklükte ayarlanan ortografik bir iz düşüm düzenlenir. Son olarak, köşe noktaları görüş alanının köşeleri ile çakışık olacak biçimde boyutlandırılan 2 boyutlu bir dikdörtgen, üzerine az önce ayarlanmış olan doku resmi kaplanacak biçimde görselleştirilmek üzere GİB’ye gönderilir. Bu sayede doku resmi üzerinde bulunan ve “doku hücresi” (*texel*) olarak da bilinen her bir piksel için GİB üzerinde ayrı bir “piksel işleyici” (*pixel processor*) çalıştırılması sağlanır.

Her bir piksel işleyici Şekil 4.12’te verilen algoritmaya göre bir doku hücresinin yakın çevresini inceleyerek doku hücresini komşuları ile gruplamaya çalışır ve sonuç bilgisini KYM renk formatında çizim belleğine yazar. Doku hücresi siyah renkte ise sadece bu durumu belirten bir bilgi, çizim belleğinde bu doku hücresinin koordinatlarının karşılık geldiği alana yazılır ve bu doku hücresi için başka hiçbir işlem yapılmaz. Eğer doku hücresi beyaz renkte ise, çevresindeki diğer doku hücrelerini değerlendirerek minimum doku hücresi kimlik numarasına sahip doku hücresini kendisine lider olarak seçer. Doku hücresi kimlik numarası (*texel ID*), doku resminin sol alt köşesindeki doku hücresi için sıfırdan başlayan ve satır öncelikli olarak diğer doku hücreleri için artarak devam eden biricik sıra numarasıdır. Doku hücresi kimlik numarasını kullanarak doku hücresinin doku resmindeki görelî koordinatlarını hesaplamak mümkün olduğu gibi doku hücresinin koordinatlarından doku hücresi kimlik numarasını belirlemek de mümkündür. Ancak ilk geçişte lider olarak seçilen doku hücresinin koordinatları hâlihazırda hesaplandığı için çizim belleğinde işlenen

doku hücresinin koordinatlarının belirttiği alana, lider doku hücresinin kimlik numarası yerine koordinatları yazılır. Böylelikle ikinci geçişte doku hücresi kimlik numarasından doku hücresi koordinatlarına dönüşüm yapmak üzere tekrardan hesaplama yapılmasının önüne geçilmiş olur.

```

Input: Image (image bound as texture)
         Atomic counter variable leader_count
         x, y (texture coordinates of current fragment)
Output: Temp (FBO bound as draw buffer)
leader_count = 0;
forall the fragment shader instances in parallel do
  if Image[x, y].red == 0 then    /* is black */
    | Temp[x, y] = (-2, 0, 0);
  end
  if Image[x-1, y-1].red == 1 then /* is white */
    | Temp[x, y] = (-1, x - 1, y - 1); /* lower left */
    | return;
  end
  if Image[x, y-1].red == 1 then  /* is white */
    | Temp[x, y] = (-1, x, y - 1);  /* lower */
    | return;
  end
  if Image[x+1, y-1].red == 1 then /* is white */
    | Temp[x, y] = (-1, x + 1, y - 1); /* lower right */
    | return;
  end
  if Image[x-1, y].red == 1 then  /* is white */
    | Temp[x, y] = (-1, x - 1, y);  /* left */
    | return;
  end
  Temp[x, y] = (leader_count, x, y); /* itself */
  leader_count = leader_count + 1;
end

```

Şekil 4.12 Beyaz piksel gruplarının merkezlerini bulmak için gerçekleştirilen ilk geçişin algoritması

Beyaz bir doku hücresi, çevresinde başka hiç beyaz doku hücresi bulunmaması durumunda kendi kendisini lider olarak kabul eder ve kendi koordinatlarını çizim belleğindeki ilgili alanına yazar. Burada değinilmesi gereken önemli bir nokta vardır. Beyaz doku hücreleri, minimum kimlik numarasına sahip doku hücrelerini lider olarak seçeceği için kendilerinden

daha büyük kimlik numarasına sahip komşu hücreleri kontrol etmelerine gerek yoktur (Şekil 4.13). Çünkü beyaz doku hücreleri kendilerinden daha küçük kimlik numarasına sahip hiçbir beyaz doku hücresiyle komşu olmamaları durumunda kendi kendilerini lider olarak kabul etmektedir. İlk geçişin sonunda aynı grupta yer alacak doku hücreleri arasında bir liderlik zinciri oluşur.

Texel ID ₆	Texel ID ₇	Texel ID ₈
Texel ID ₃	Texel ID ₄	Texel ID ₅
Texel ID ₀	Texel ID ₁	Texel ID ₂

Şekil 4.13 Mavi renk ile gösterilen beyaz doku hücresinin işlendiği varsayıldığında sadece kendisinden daha küçük doku hücresi kimlik numarasına sahip olabilecek turuncu renk ile gösterilmiş doku hücrelerinin kontrol edilmesi

Belirtilmesi gereken bir diğer önemli nokta ise, beyaz bir doku hücresi kendi kendisini lider olarak belirlediğinde, tespit edilen grupların (lider beyaz doku hücrelerinin) sayısını tutan ve her yeni birey oluşturma sürecinde sıfırlanan bir atomik sayacın değeri bir artırılır. Atomik sayaç [134], okuma ve okuma-değiştirme-yazma işlemlerinin gerçekleştirilebildiği ve “atomik” isminin gerektirdiği bölünmezliği sağlamak üzere bir işlem bitmeden paralel diğer hiçbir işlemin işletiminin başlatılmadığı bir GLSL değişken türüdür. Atomik sayacın bu özelliklerinden faydalanılarak paralel bir işletim yapılmasına rağmen her bir lider doku hücresine biricik bir sıra numarası atanabilmektedir. Bu sıra numaraları grup bilgilerini taşıyacak farklı lider doku hücrelerini temsil eder. Ayrıca bu numaralar ikinci geçişte, her grupta yer alacak doku hücrelerinin bilgilerinin toplanarak, çıktı olarak kullanılacak bellek alanının en başından başlayacak şekilde bu sıra numaralarına göre yerleştirilmelerine yardımcı olacaktır.

Haciomeroglu ve arkadaşları [73], GİB tabanlı kümeleme algoritmalarının ikinci geçişinde kümelenmiş bireylerin liderlik zincirlerini çözümlmek ve küme üyelerinin pozisyon, hız gibi özelliklerini bir araya toplamak üzere yeni bir yaklaşım geliştirmişlerdir. Küme üyelerinin bilgilerini taşıyan “köşe” (*vertex*) ilkelerini, küme liderleri ile aynı pozisyonda olacak biçimde oluşturmak ve “yayınlamak” (*emit*) için GİB’nin “geometri işleyici” (*geometry processor*) donanımlarını kullanmışlardır. Bunun yanı sıra, GİB’nin “karıştırma” (*blending*) [135] özelliğinden de faydalanarak aynı pozisyonda yer alan küme üyelerinin toplam sayısının, pozisyonlarının toplamının ve hız vektörlerinin toplamının hesaplamasını ve çıktı belleği üzerinde küme lideri birey için ayrılmış alana yazılmasını sağlamışlardır.

Beyaz pikselleri gruplamak için kullanılan ikinci geçiş için Haciomeroglu ve arkadaşları [73] tarafından geliştirilen bu yaklaşım bazı değişiklikler yapılarak kullanılmıştır. Her bir doku hücresinin liderlik zincirlerini, en uçta yer alan asıl liderlerini buluncaya kadar izlemek ve bu doku hücresini asıl liderine bağlamak üzere geometri işleyiciler üzerinde çalışacak, GLSL tabanlı benzer bir algoritma geliştirilmiştir. Farklı olarak burada kullanılan geometri işleyiciler, doku hücrelerinin bilgilerinin yazılacağı köşe ilkelerinin pozisyonlarını, ilk geçişte lider doku hücrelerine atanan biricik sıra numaralarını kullanarak hesaplamaktadır. GİB’nin karıştırma özelliğinin birebir toplama fonksiyonunda kullanılmasıyla birlikte her grup için grubu oluşturan doku hücrelerinin sayısı ve koordinatlarının toplamı hesaplanarak çıktı belleği üzerinde en baştan başlamak suretiyle grup sıra numarasına göre ilgili alanlara yazılabilmektedir. İkinci geçişin algoritması Şekil 4.14’te verilmiştir.

```

Input: Temp (output image of the previous pass)
          x, y (texture coordinates of current primitive)
Output: Image (FBO bound as draw buffer)
forall the geometry shader instances in parallel do
  if Temp[x, y].red == -2 then    /* is black */
  |   Discard vertex;
  end
  else
  |   xorg = x;
  |   yorg = y;
  |   while Temp[x, y].red == -1 do
  |   |   xtemp = Temp[x, y].green;
  |   |   ytemp = Temp[x, y].blue;
  |   |   x = xtemp;
  |   |   y = ytemp;
  |   end
  |   texel_pos = Calculate current texel's position
  |   using xorg and yorg;
  |   leader_pos = Calculate leader's vertex
  |   position using Temp[x, y].red;
  |   vertex.color = (1, texel_pos.x, texel_pos.y);
  |   vertex.pos = (leader_pos.x, leader_pos.y);
  |   emit(vertex);
  end
end

```

Şekil 4.14 Beyaz piksel gruplarının merkezlerini bulmak için gerçekleştirilen ikinci geçişin algoritması

Her iki geçiş de tamamlandıktan sonra atomik sayaç belirlenen farklı grupların sayısını tutmaktadır. Grup sayısının bilinmesi sayesinde çıktı belleğinin tamamı yerine, en başından başlayarak belirlenen grupların bilgilerinin yazılı olduğu ilgili hücreleri içeren küçük bir bölümünün GİB belleğinden AİB belleğine transfer edilmesi yeterli olacaktır. Sonrasında ise her bir bellek hücresinde yer alan bilgiler kullanılarak grubu oluşturan doku hücrelerinin koordinatları toplamının doku hücresi sayısına bölünmesi suretiyle grup merkezleri basit bir biçimde hesaplanabilir. Nihayetinde elde edilen grup merkezleri önceki bölümlerde anlatıldığı gibi yeni birey oluşturmak amacıyla çözüm olarak kullanılır. Bu grup merkezleri arasından yalnızca biri yeni birey oluşturmada çözüm olarak kullanılacaktır. Çözüm olarak kullanılacak grup merkezinin seçilmesini sağlamak üzere grubu oluşturan birey sayısına bakılarak en büyük ve en küçük grubun seçilmesi mümkün olmakla birlikte grup merkezleri arasından tamamen rastgele bir seçim yapılması da mümkündür. Bu seçim yöntemlerinin

gerçek zamanlı benzetimlerde ortalama ve maksimum birey sayılarına etkilerini arařtıran deneyler ve sonuçlarına “6.1.1 Grup Merkezi Seçim Yöntemlerinin İrdelenmesi” bölümünde detaylıca yer verilmiştir.

Burada belirtilmesi gerek bir diđer önemli nokta ise, kullanılan iki geçişli gruplama algoritmasının alt grupları birleştirerek tek bir bağlantılı bileşen oluşturmada tam olarak başarılı olamayabileceğidir. Daha önce de belirtildiği gibi bu durumun nedeni paralel BBE tekniklerinin bağlantılı bileşenleri tamamen birleştirebilmesi için çok sayıda geçiş gerektirmesidir. Ancak belirlenecek tüm grup merkezleri birer olası çözüme karşılık geleceğinden, bu sorun burada geliştirilen yaklaşım için herhangi bir engel, eksiklik veya sakınca teşkil etmemektedir.

4.4. Benzetimde Yer Alan Bireylerin Hareket Ettirilmesi

Geliştirilen GİB tabanlı TYPÇY yönteminde de bireyler tek bir doğrusal ve sabit hızlı yörünge parçasından oluşan çarpışmasız yollara sahip olabilmektedir. Bireylerin benzetime girişleri esnasında kendilerine tahsis edilen bu TYPÇY’ler üzerinde belirlenen sabit hız vektörleri ile herhangi bir doğrultu veya hız değişikliği yapmadan benzetim boyunca yürümlerinin hiçbir çarpışmaya neden olmayacağı da garanti edilmiştir. Bu nedenle bireyler için benzetimde yer aldıkları süre boyunca çarpışma tespiti veya çarpışmadan kaçınma manevraları için herhangi bir işlem yapılmasına ihtiyaç yoktur.

Bireylerin hareket ettirilmesi amacıyla her bir benzetim zaman adımında, her bir bireyin güncel pozisyonları tıpkı AİB tabanlı TYPÇY yönteminde olduğu gibi Eşitlik (3.1)’deki formüle göre hesaplanır. Ancak AİB tabanlı TYPÇY yönteminin aksine, GİB tabanlı TYPÇY metodunun birey oluşturma sürecinde, bireylerin benzetim alanı üzerinde konum-zamansal dizinlemesine ihtiyaç duyulmamaktadır. Dolayısıyla ne böyle bir dizinleme için herhangi bir veri yapısına ne de AİB tabanlı TYPÇY yönteminde olduğu gibi her bir benzetim zaman adımında ziyaret ettiği hücreden tamamen çıkan bireyler nedeniyle bu dizinleme yapısının güncellenmesine gerek vardır.

Pozisyon güncellemesinin hemen sonrasında, yine AİB tabanlı TYPÇY yönteminde kullanılan ve Eşitlik (3.2)’de verilen formüle göre her bir bireyin bitiş noktasına ulaşip ulaşmadığı kontrol edilir. Böylelikle bireylerin sahip oldukları çarpışmasız yolların bitiş noktalarına ulaştıkları ($\alpha = 90^\circ$) ya da daha ötesine geçtikleri ($\alpha > 90^\circ$) tespit edilerek benzetimden çıkarılmaları sağlanır ve benzetimin devamında bitiş noktalarının üzerinde yer aldığı benzetim alanı kenarının ötesine geçerek benzetim alanının dışına çıkmaları engellenir.

5. ÇOK YÖRÜNGE PARÇALI ÇARPIŞMASIZ YOL YÖNTEMİ

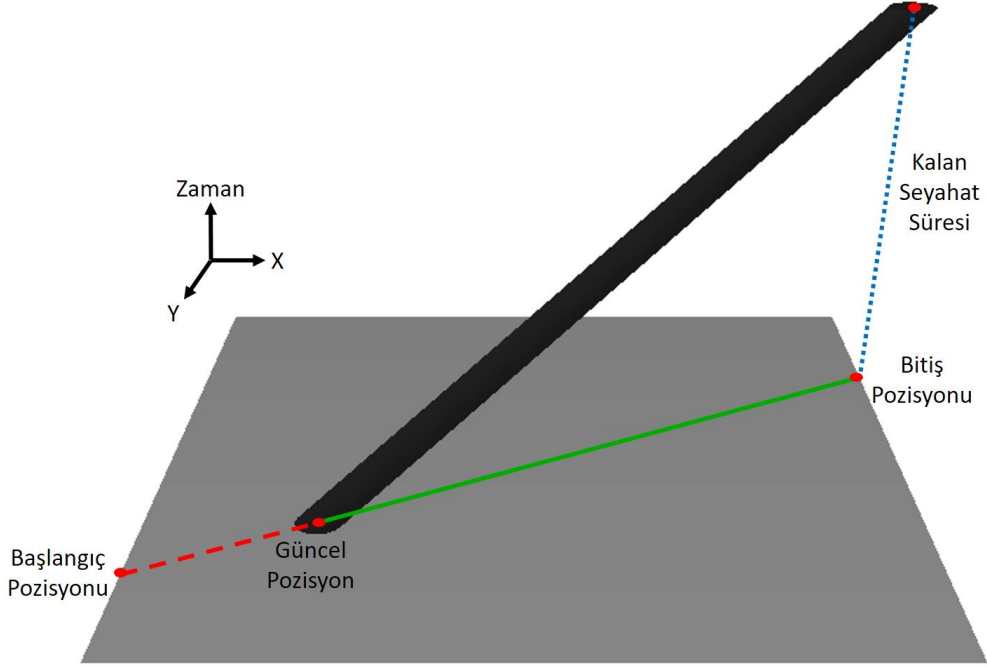
Önceki bölümde detayları verilen GİB tabanlı TYPÇY kullanımına dayalı yönlendirmesiz gezinim konseptinin daha da geliştirilme potansiyeli oldukça yüksektir. Buradan yola çıkılarak tez kapsamında GİB tabanlı TYPÇY yöntemini temel alan bir diğer yönlendirmesiz gezinim yöntemi önerilmiştir. Geliştirilen ÇYPÇY yönteminde, bireylerin benzetim alanı üzerinde hareket ettirilmesini sağlamak amacıyla sadece TYPÇY'ler kullanmak yerine birden fazla doğrusal ve sabit hızlı yörüngeye uç uca eklenerek başlangıç noktasından bitiş noktasına uzanan çarpışmasız yollar oluşturulması öngörülmüştür. Yörünge parçaları belirlenirken yeni eklenecek yörünge parçasının kendisinden önceki parçaya göre ön tanımlı limitler çerçevesinde doğrultu ve/veya hız değişiklikleri yapmasına olanak tanınmıştır. Geliştirilen bu gezinim yöntemini de yeni birey oluşturma ve bireylerin hareket ettirilmesi bileşenlerine ayırmak ve bu alt başlıklar altında incelemek yerinde olacaktır.

5.1. Yeni Birey Oluşturulması

Geliştirilen bu gezinim yöntemi, 2 boyutlu bir benzetim alanını yürüten insan bireyler ile doldurmak üzere, yine GİB'den faydalanan bir birey oluşturma süreci kullanmaktadır. Geliştirilen diğer gezinim yöntemlerine benzer biçimde bireyler 2 boyutlu benzetim alanı üzerinde r yarıçapına sahip diskler ile ifade edilmektedir. Bu benzetim alanına dik olan üçüncü boyut ise zaman boyutu olarak kabul edilmiştir. GİB tabanlı TYPÇY yönteminde olduğu gibi bireylerin benzetim alanının bir kenarı üzerinden başlayıp bu kenarın tam karşısında yer alan kenara ulaşacak biçimde seyahat ettiğini ve yine bireylerin ön tanımlı $[s_{min}, s_{max}]$ aralığında yürüme hızı değerlerine sahip olabileceğini hatırlatmakta fayda vardır.

Önceki bölümde de belirtildiği üzere 2 boyutlu benzetim alanı üzerinde sabit hızlar ile doğrusal bir yörüngeyi takip ederek hareket eden diskler, 3 boyutlu konum-zaman uzayında doğrusal ve silindirik borulara karşılık gelmektedir. Şekil 5.1'de tek bir doğrusal ve sabit hızlı yörünge parçasına sahip olan bir bireyin başlangıç pozisyonundan bitiş pozisyonuna olan seyahatinin bir anında 3 boyutlu konum-zaman uzayında kapladığı silindirik KZYE hacmi gösterilmektedir. Benzetimin herhangi bir anında, 3 boyutlu konum-zamansal uzay, benzetimi oluşturan bireylere ait olan bu borular ile dolu olacaktır. Yeni bir birey oluşturmak için ise bu yeni bireyin yörünge parçalarının karşılık geleceği benzer borular için var olan bu boru kümesi içerisinde uygun bir boşluk bulunması gerekmektedir. Bu boşlukların bulunması amacıyla GİB tabanlı TYPÇY yöntemine benzer biçimde, GİB'nin hızlı görselleştirme iş hattından faydalanılarak boru kümesinin iz düşüm resimleri yakalanmakta

ve bu resimler üzerinde yeni bireylerin borularını yerleřtirmek üzere yeterli büyüklükte boşluklar aranmaktadır.



Şekil 5.1 Tek bir doğrusal ve sabit hızlı yörünge parçasına sahip olan bir bireyin 3 boyutlu konum-zaman uzayında kapladığı silindirik hacim

Parçalı yol oluşturabilmek amacıyla ilk önce herhangi bir bireye ait çarpışmasız yolun sahip olabileceği yörünge parçası sayısını sınırlayan “maksimum parça sayısı” (*max_num_of_segments*) isimli bir parametre tanımlanmıştır. Buna göre, her bir yol $[I, max_num_of_segments]$ kapalı aralığında yörünge parçasına sahip olabilmektedir. Bu nedenle yeni birey oluşturma süreci, yeni bireyin yolu başlangıç kenarının tam karşısındaki kenara ulaşmaya kadar ya da yeni bireyin inşası devam eden yolunun sahip olduğu yörünge parçası sayısı bir noktada *max_num_of_segments* sınırını aşmaya kadar devam eder.

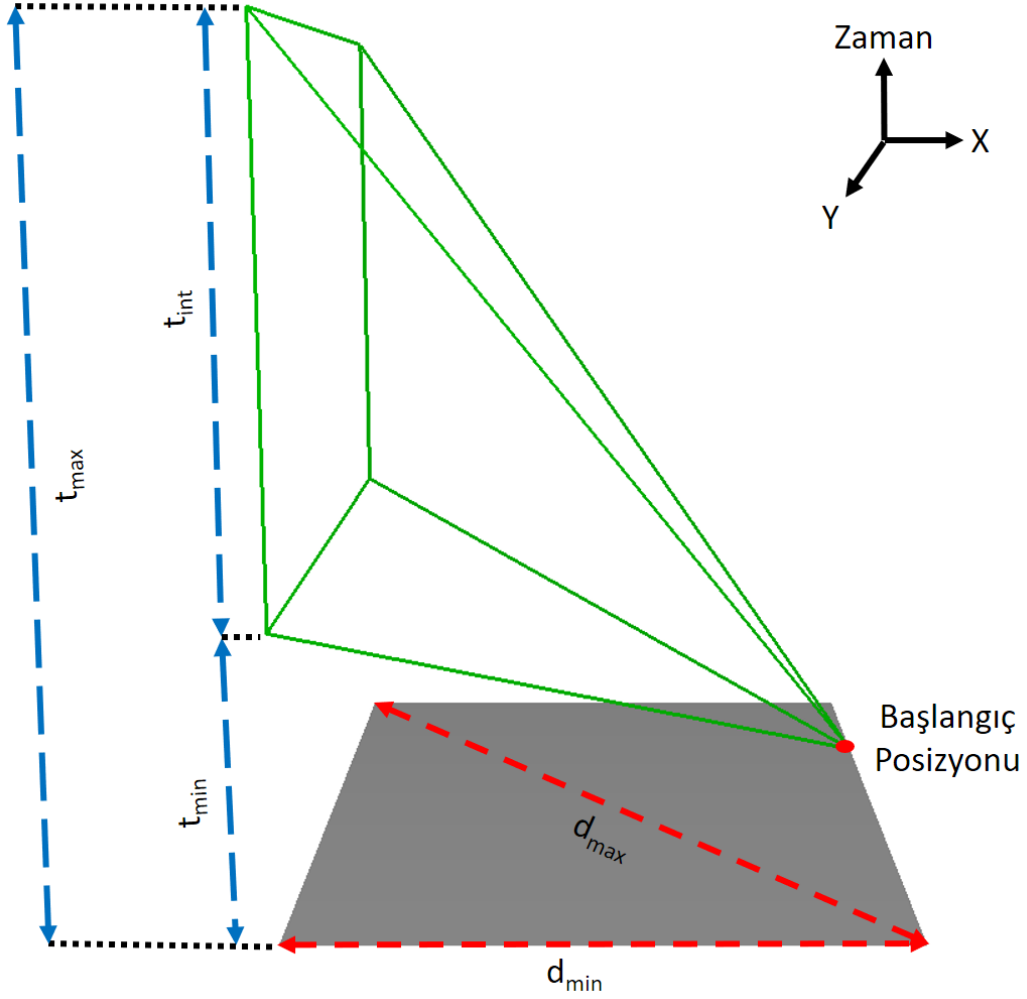
Yeni bir bireyin oluşturulması, benzetim alanının rastgele seçilen bir kenarı üzerinde yer alan rastgele bir başlangıç pozisyonu seçilmesiyle başlar. Başlangıçta yeni bireyin çarpışmasız yolunun güncel “bitim noktası” (*endpoint*), bu başlangıç pozisyonuna karşılık gelir. Sonrasında birbirini izleyen yörünge parçası arama yinelemeleri yapılarak benzetim alanının karşı kenarına ulaşacak çarpışmasız bir yol oluşturmaya çalışılır. Her yörünge parçası arama yinelemesi, yolun güncel bitim noktasından başlayacak şekilde yapılmakla

birlikte kameranın ayarlanması, konum-zamansal görselleştirme ve çözüm analizi alt adımlarından oluşur.

5.1.1. Kameranın Ayarlanması

Her yörünge parçası arama yinelemesinde perspektif görüşe sahip bir kameranın bakış noktası, oluşturulan yolun konum-zamansal uzayda yer alan güncel bitim noktasıyla çakışacak biçimde yerleştirilir. Burada akılda bulundurulması gereken önemli bir husus, ilk yörünge parçası arama yinelemesinde konum-zamansal yolun güncel bitim noktası, 2 boyutlu benzetim alanına karşılık gelen düzlemin tam üzerinde yer alacakken sonraki yinelemelerde yolun güncel bitim noktasının, zaman ekseninde bu düzlemin yukarısında yer alacak olmasıdır. Yolun güncel bitim noktasına yerleştirilecek kamera, başlangıç kenarının karşısındaki kenara dik bir doğrultu ile bakacak bir biçimde yönlendirilerek görüş hacminin yakın ve uzak düzlemlerinin karşı kenar ile paralel ve aynı zamanda benzetim alanı ile dik olması sağlanır. Kamera görüş hacminin ayarlamaları da GİB tabanlı TYPÇY yönteminde kullanılan kameraya benzer biçimde yapılır. Görüş hacminin yakın düzlemi, kameranın bakış noktasına olabildiğince yakın yerleştirilirken (0.001 m) uzak düzlemi, karşı kenardan bir miktar daha ötede olacak biçimde yerleştirilir. Kameranın yatay görüş açısı, uzak düzlem karşı kenarı tamamen kapsayacak biçimde ayarlanır. Diğer taraftan, kameranın dikey görüş açısı, uzak düzlemin yüksekliği yaklaşık olarak “seyahat süre aralığı” (t_{int}) kadar olacak biçimde belirlenir. Bu arada seyahat süre aralığının, benzetim alanının karşılıklı kenarları arasındaki uzaklığı (d_{min}) ve köşegen uzunluğu (d_{max}) gibi ölçülerinin yanı sıra ön tanımlı minimum (s_{min}) ve maksimum (s_{max}) yürüme hızları kullanılarak Eşitlik (5.1)’deki gibi hesaplandığını belirtmek gerekir. Daha sonra görüş hacmi uygun bir biçimde dikey ve yatay eksenlerde eğilerek uzak düzlemin alt kenarının, kameranın bakış noktasından minimum seyahat süresi kadar yüksekte olması sağlanırken benzetim alanının karşı kenarı ile aynı hizada olması sağlanır (Şekil 5.2). Bu sayede olası bitiş pozisyonlarının ve yürüme hızlarının tamamını kapsayan bir görüş oluşturulması garantilenmiş olur.

$$t_{int} = t_{max} - t_{min} = \frac{d_{max}}{s_{min}} - \frac{d_{min}}{s_{max}} \quad (5.1)$$



Şekil 5.2 Konum-zamansal görselleştirme öncesi perspektif kameranın ayarlanması

5.1.2. Bireylerin Konum-Zamansal Görselleştirilmesi

Kameranın ayarlanmasından sonra sıra, benzetimi oluşturan bireylerin konum-zamansal borularının görselleştirilerek kameranın bakış noktasından bir iz düşüm resmi elde edilmesine gelir. Ekran dışı olarak gerçekleştirilecek bu görselleştirmenin sonuçlarının yazılacağı bir RKAN, benzetimin en başında henüz hiçbir birey oluşturulmadan önce tanımlanır. Görselleştirme sonunda her pikselin renk ve derinlik bilgisinin saklanması amacıyla bu RKAN'ye bağlı bir "renk belleği" (*color buffer*) ve bir "derinlik belleği" (*depth buffer*) oluşturulur. Her iki bellek de kameranın uzak düzleminin boyutlarıyla orantılı olacak biçimde $w_{env} * k$ genişliğine ve $t_{int} * k$ yüksekliğine sahip olacak şekilde ayarlanır. Renk belleğinin her bir hücresi (piksel) kayan noktalı sayı formatında KYM bileşenlerinden

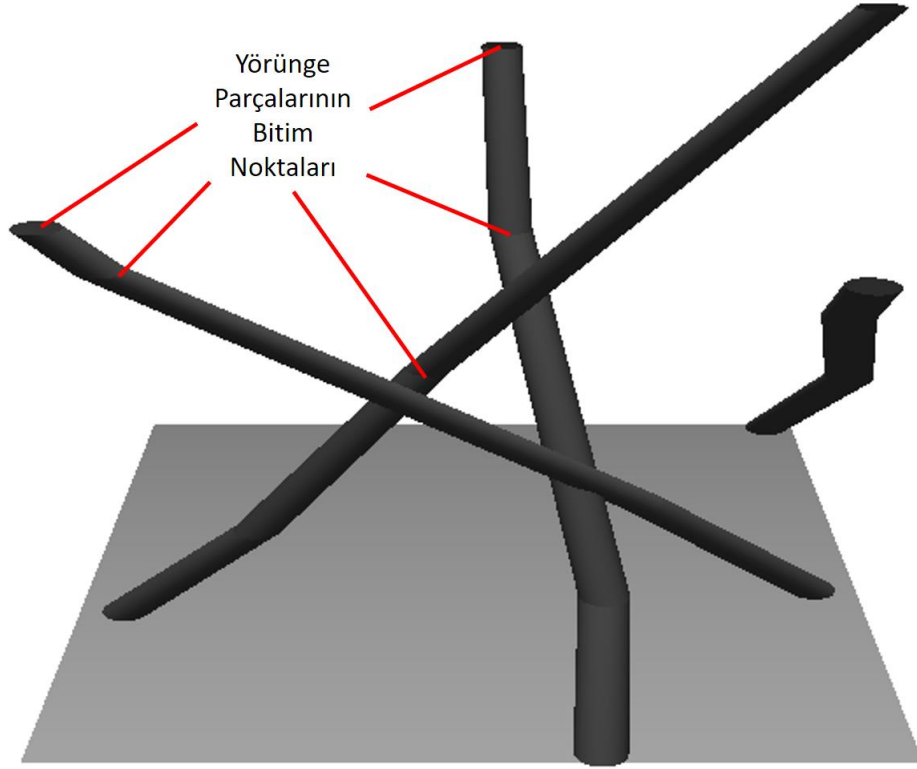
meydana gelirken derinlik belleğinin her bir elemanın kayan noktalı sayı formatında derinlik bilgisine karşılık gelen tek bir bileşenden oluştuğunu belirtmekte fayda vardır.

Bireylerin görselleştirilmesi işleminin algoritması aşağıda verilmiştir.

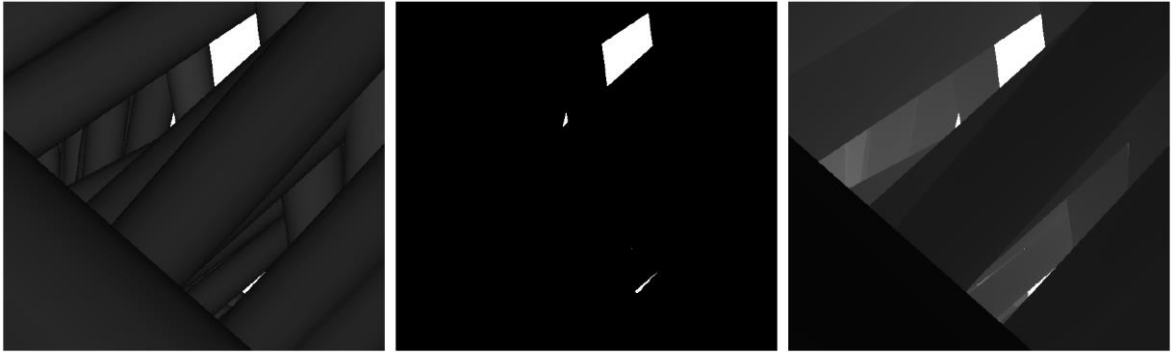
- Öncelikle sahne arka plan rengi beyaz olarak ayarlanır.
- Görselleştirmeden hemen önce GİB'nin “derinlik testi” (*depth test*) [136] özelliği varsayılan davranışı olan “daha az” (*less*) fonksiyonu kullanılarak aktif hale getirilir.
- Benzetimde var olan bireylerin konum-zamansal görselleştirilmesi için siyah renkli, r yarıçapına ve birim yüksekliğe sahip bir silindir modeli kullanılır.
- Her bir bireyin sahip olduğu çarpışmasız yolu oluşturan her bir yörünge parçası için:
 - Silindir modeli, tabanının merkez noktası, yörünge parçasının 3 boyutlu konum-zaman uzayındaki başlangıç pozisyonu ile çakışık olacak bir biçimde 3 boyutlu konum-zaman uzayı içerisinde tabanı üzerinde dik olarak konumlandırılır.
 - Silindir, X ve Z boyutlarında, yarıçapı $2 * r + \varepsilon$ olacak şekilde ölçeklenir.
 - Silindir modelinin yüksekliği; tavanının merkez noktası, ilgili yörünge parçasının 3 boyutlu konum-zaman uzayındaki bitiş pozisyonu ile aynı yükseklikte olacak biçimde, bireyin yörünge parçası üzerindeki seyahat süresi kadar ölçeklenir.
 - Silindir modeli; tavanının merkez noktası, ilgili yörünge parçasının 3 boyutlu konum-zaman uzayındaki bitiş pozisyonu ile çakışık olacak biçimde eğilerek, yörünge parçasının hız vektörüne bağlı olarak başlangıç pozisyonundan bitiş pozisyonuna doğru eğimli silindirik bir boru elde edilir.

Her bir yörünge parçası için 3 boyutlu silindirik bir boru çizilmesi sonucu Şekil 5.3'te gösterildiği gibi konum-zaman uzayı içerisinde kırıklı borular biçiminde KZYE'ler oluşur. “daha az” fonksiyonunu kullanan derinlik testine göre, sadece derinlik belleği üzerinde karşılık geldiği alanda kayıtlı derinlik değerinden daha az derinlik değerine sahip olan pikseller derinlik testini geçmiş sayılır. Derinlik testini geçen bu gibi pikseller, derinlik ve renk bilgilerini derinlik ve renk belleklerinde karşılık geldikleri alanlara yazmaya hak kazanır. Derinlik testini geçemeyen pikseller ise yok sayılarak derinlik ve renk bellekleri üzerinde değişiklik yapmalarına izin verilmez. Dolayısıyla görselleştirme işlemi tamamlandıktan sonra; renk belleği, derinlik testi kullanılmadığı varsayıldığında elde

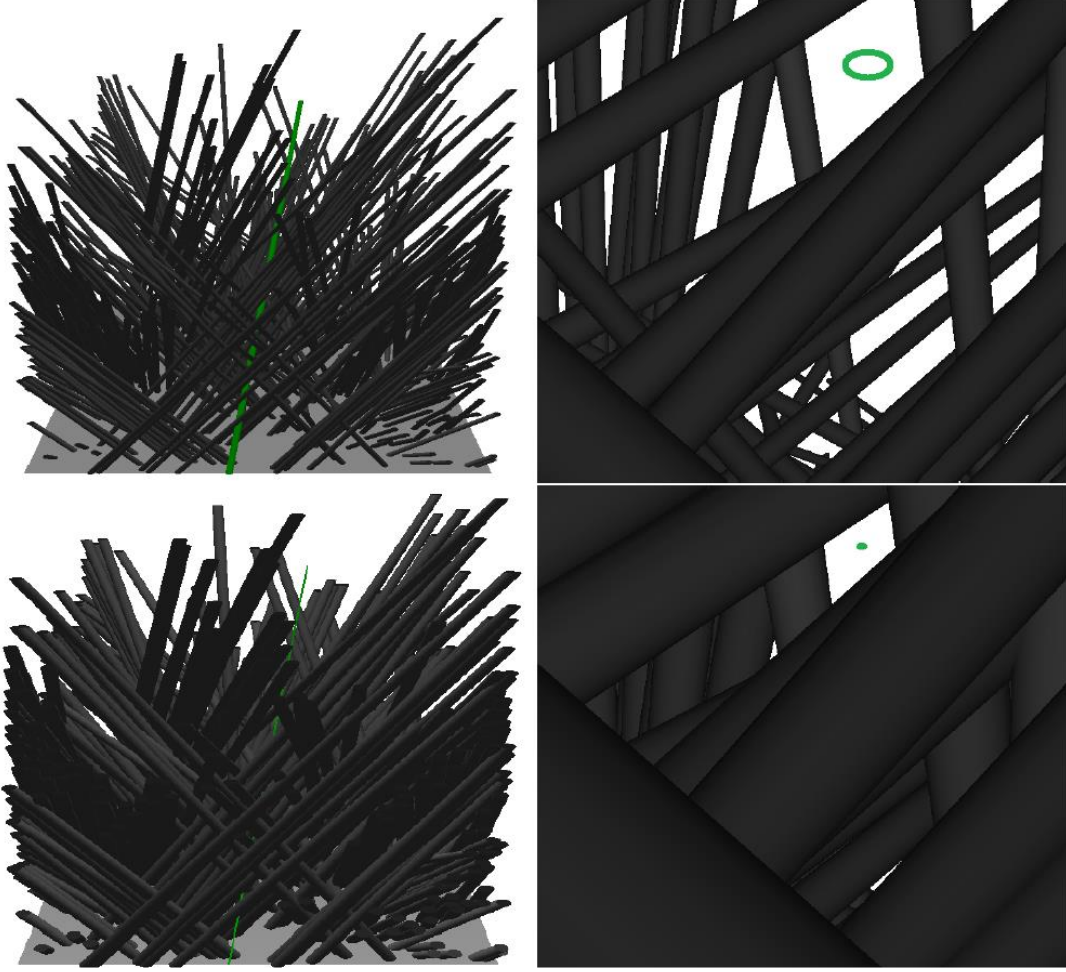
edilecek iz düşüm resmi ile tıpatıp aynı resmi içeriyorken derinlik belleği, bu iz düşüm resmi üzerindeki piksellerin her birinin derinlik değerlerini içeriyor olacaktır (Şekil 5.4).



Şekil 5.3 Her bir yolun sahip olduğu her bir yörünge parçası için oluşturulan konum-zamansal hacimler



Şekil 5.4 (Sol) Kameranın bakış açısından silindirik borulardan oluşan konum-zaman uzayının görünümü, (Orta) Kameranın bakış açısından elde edilmiş siyah beyaz iz düşüm resmi, (Sağ) Derinlik belleği içeriğinin doğrusallaştırılmış ve normleştirilmiş derinlik değerlerine göre oluşturulmuş monokrom resmi



Şekil 5.5 (Üst-Sol) Var olan bireylerin borularının orijinal yarıçapları ile görselleştirilmesi (Üst-Sağ) durumunda yeni birey oluşturmak için gereken yeterli genişlikte bitişik beyaz piksel kümesi bulmak gerekir. (Alt-Sol) Var olan bireylerin borularının orijinal yarıçaplarının iki katı ile görselleştirilmesi sonucu yeni bireyi sadece bir doğru ile ifade etmek (Alt-Sağ) ve tek bir beyaz pikselin çarpışmasız bir çözüm olduğunu garanti etmek mümkündür.

Benzetimi oluşturan bireylerin yukarıda verilen algoritmaya göre görselleştirilmesinden sonra ayarlanmış olan kameranın bakış açısından sahnenin ekran dışı bir iz düşüm resmi yakalanır. Bu resmin siyah pikselleri var olan bireylerin konum-zamansal borularına karşılık gelirken beyaz pikselleri karşı kenara açılan boş oyuklara karşılık gelmektedir. GİB tabanlı TYPÇY yönteminde olduğu gibi burada da görselleştirme esnasında, bireylerin konum-zamansal borularının yarıçapı iki katına çıkarılarak yeni bireyi temsil eden borunun genişliğinin en baştan hesaba katılması sağlanmıştır. Bu sayede iz düşüm resmi üzerinde yeni bireyi temsil eden borunun sığabileceği genişlikte beyaz boşluklar aramak yerine

konum-zaman uzayında yeni bireyi sadece hacimsiz bir doğru olarak ifade etmek ve çarpışmasız bir çözüm olacağı garanti edilen tek bir beyaz piksel seçmek mümkündür (Şekil 5.5).

5.1.3. Olası Çözümlerin Analiz Edilmesi

Yeni birey için oluşturulacak yolun bir sonraki yörünge parçasını belirlemek üzere renk ve derinlik belleklerine kaydedilmiş görselleştirme çıktısının analiz edilerek, çözüm olarak kullanılacak piksele karar verilmesi gerekmektedir. Renk belleğinde yer alan resim içerisinde beyaz piksellerin var olması yolun güncel bitim noktasından başlayıp karşı kenara kadar ulaşan en az bir doğrusal, sabit hızlı ve çarpışmasız yörünge parçası bulunduğu anlamına gelmektedir. Her piksel farklı bir çözüme karşılık geldiği üzere her piksel için bu özelliklere sahip farklı bir yörünge parçası tanımlanması mümkündür. Yolun en başında ilk yörünge parçası aranırken karşılaşılabilecek bu beyaz pikseller, başlangıç pozisyonundan karşı kenara kadar uzanan doğrusal ve sabit hızlı yörünge parçalarına karşılık gelecektir. Bu durumda beyaz piksellerin karşılık geldiği olası çözümlerden birine karar verilerek önceki bölümde anlatılan GİB tabanlı TYPÇY yönteminde olduğu gibi başlangıç noktasından başlayıp karşı kenara kadar uzanan tek bir doğrusal ve sabit hızlı yörünge parçasından meydana gelecek çarpışmasız bütün bir yol oluşturmak mümkündür. Bu nedenle ilk yörünge parçası arama yinelemesinde herhangi bir beyaz piksel ile karşılaşırsa uygun bir tanesi çözüm olarak seçilerek yeni oluşturulacak bireyin yolu olarak atanır ve yeni bireyin oluşturulması için tüm gereksinimler karşılandığı üzere yörünge parçası arama yinelemeleri bu noktada sonlandırılır. Beyaz piksele karar verilmesinden sonra yeni bireyin bitiş pozisyonu ve yürüme hızının belirlenmesi oldukça kolaydır. İz düşüm resminin, kameranın görüş hacminin uzak düzlemi üzerinde birebir örtüşecek biçimde kaplı olduğu varsayıldığında, seçilen beyaz pikselin benzetim alanı üzerine iz düşümü karşı kenar üzerinde yer alan bitiş pozisyonuna karşılık gelir. Benzer biçimde, başlangıç pozisyonundan başlayıp seçilen bu beyaz piksele ulaşan hayali doğrunun benzetim alanına göre eğiminin çarpma işlemine göre tersi de yürüme hızına karşılık gelir.

Herhangi bir yörünge parçası arama yinelemesinde renk belleği içerisinde hiçbir beyaz pikselin bulunmaması durumunda geliştirilen bu yeni yöntem o ana kadar üretilen yoldan hemen vazgeçmeyerek yolun inşasını devam ettirmek üzere bitim noktası karşı kenara ulaşamasa bile yeni bir yörünge parçası oluşturmayı deneyecektir. Bu gibi durumlarda en mantıklı seçenek, bitim noktası karşı kenara en yakın olan yörünge parçası ile devam edilmesidir. Çünkü yeni bireyler için üretilecek her yol zaten karşı kenara ulaşmaya

çalışmaktadır. Bu seçim kararının arkasında yatan mantık ise şudur: Her yörünge parçası arama yinelemesinde karşı kenara mümkün olan en fazla miktarda yaklaşmak için çabalamak, karşı kenara doğru daha uzun mesafe kateden yörünge parçaları oluşturulmasını sağlayacak ve dolayısıyla karşı kenara ulaşmak için gerekecek maliyetli yörünge parçası arama yinelemelerinin sayısını azaltacaktır. Bu nedenle ÇYPÇY yönteminin birey oluşturma süreci, ardışık yörünge parçası arama yinelemelerinde karşı kenara en yakın bitim noktasına sahip yörünge parçalarını tercih ederek yerel olarak en iyi kararı veren “açgözlü” (*greedy*) bir yaklaşım kullanmaktadır.

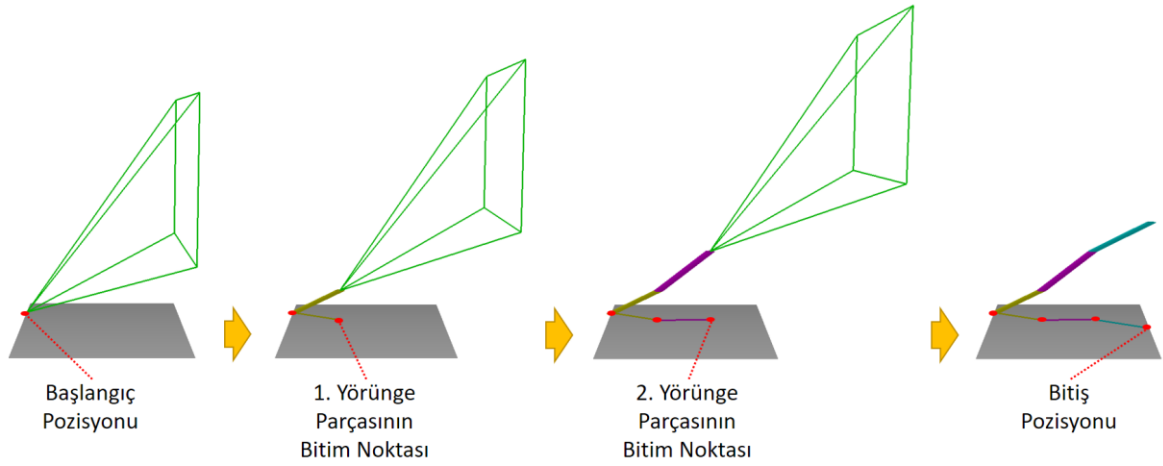
Bu amaçla piksellerin derinlik belleği içerisinde yer alan derinlik bilgisi kullanılarak kameranın görüş hacminin yakın düzleminden en uzak, dolayısıyla benzetim alanının karşı kenarına en yakın siyah pikselin belirlenmesi sağlanmıştır. Belirlenen bu siyah pikseli çözüm olarak kullanıp beyaz pikseller için yukarıda anlatılan hesaplamaların birebir aynısını gerçekleştirerek doğrusal bir yörünge parçası ve sabit bir yürüme hızı belirlenmesi halen mümkündür. Ancak daha önce de bahsedildiği üzere siyah piksellerden herhangi biri kullanılarak üretilen sabit hız vektörü ile yürümek, eninde sonunda benzetimde var olan bireylerden en az biri ile çarpışmaya neden olacaktır. Bu nedenle siyah piksel kullanılarak oluşturulacak bu yörünge parçasının konum-zamansal uzaydaki bitim noktası, siyah pikselin konum-zaman uzayındaki (görüş hacmi içerisindeki) pozisyonu ile çakışık olacak biçimde bitim ucu kısmından kırılarak yörünge parçasının çarpışmasız olması garanti edilir. Bu kırılma işleminin sonucunda, yörünge parçasının yeni bitim noktasının 2 boyutlu benzetim alanına iz düşümü, kameranın görüş hacminin yakın düzleminden, kullanılan siyah pikselin derinlik değeri kadar uzakta yer alacaktır. Yine de bitim noktasının konum-zaman uzayında bir engel (KZYE) olarak düşünülen konum-zamansal bir boruya temas edecek ölçüde yakın olması, bir sonraki yörünge parçası arama yinelemesinde bu bitim noktası üzerine konumlandırılacak kameranın görüş alanının bu engel tarafından önemli ölçüde kapanmasına neden olacaktır. Kameranın çok kısıtlı bir net görüşe sahip olması ise bulunabilecek olası çözümlerin sayısını oldukça sınırlı bir hale getirecektir. Söz konusu durumdan kaçınmak adına siyah pikseller kullanılarak oluşturulan bu gibi ara yörünge parçalarının bitim noktaları, 2 boyutlu benzetim düzlemi üzerine iz düşümü yapılan yörünge parçası doğrultusu boyunca $margin_coef * r$ kadar geri çekilecek biçimde kırılır (Şekil 5.6). Burada kullanılan pay katsayısı ($margin_coef$) kullanıcı tarafından değer atanabilen ve benzetim boyunca sabit tutulan bir katsayı iken r daha önce de belirtildiği gibi bireyin yarıçap değeridir. Böylelikle sonraki yörünge parçası arama yinelemesinde kullanılacak

kameranın, bu KZYE'den bir miktar uzaklaştırılarak daha geniş bir net görüş alanına sahip olması ve dolayısıyla bulunabilecek olası çözüm sayısının artırılması sağlanır.



Şekil 5.6 Siyah pikselin derinlik değeri ve pay katsayısı kullanılarak yörünge parçasının bitim noktasının belirlenmesi

Ardışık yörünge parçası arama yinelemeleri esnasında her yörünge parçası, özel olarak tanımlanmış açı değişimi (*delta_angle*) ve hız değişimi (*delta_speed*) parametrelerinin belirlediği sınırlar dâhilinde, kendisinden önceki yörünge parçasından farklı doğrultu ve hıza sahip olma esnekliğine sahiptir. Burada kullanılan *delta_angle* parametresi; bir yörünge parçasının 2 boyutlu benzetim alanı üzerindeki yürüme doğrultusunu, kendisinden önceki yörünge parçasının doğrultusuna göre olarak değiştirebileceği maksimum açının derece cinsinden değeridir. Benzer biçimde, *delta_speed* parametresi ise bir yörünge parçasının kendisinden önceki yörünge parçasının yürüme hızına göre kendi yürüme hızını artırabileceği veya azaltabileceği maksimum hız miktarıdır. Ardışık yörünge parçaları arası geçişlerde doğrultu ve hız değişikliklerine izin verilmiş olmasına rağmen yürüme doğrultu ve hızlarının geçiş noktalarından itibaren yörünge parçaları boyunca yine sabit olduğunu hatırlatmakta fayda vardır. Birden çok yörünge parçasından oluşan ve bu yörünge parçaları arasındaki geçişlerde doğrultu ve hız değişiklikleri barındıran örnek bir çarpışmasız yolun adım adım oluşturulmasının gösterimi Şekil 5.7'de verilmiştir.



Şekil 5.7 Birden çok yörünge parçasından oluşan ve bu yörünge parçaları arasındaki geçişlerde doğrultu ve hız değişiklikleri barındıran örnek bir çarpışmasız yolun adım adım oluşturulması

Her bir çarpışmasız yolun başlangıç pozisyonundan başlayarak sıfır ve daha fazla ara yörünge parçasına (karşı kenara ulaşamayan) sahip olabileceğini belirtmek gerekmektedir. Buna ek olarak, her bir çarpışmasız yol karşı kenara ulaşmayı başarabilen yalnızca bir yörünge parçasına sahip olmalıdır ve bu yörünge parçası, yolu oluşturan son parça olmalıdır. Ancak yolu oluşturan yörünge parçaları içerisinde yalnızca ilk parça haricindekiler için doğrultu ve hız değişiklikleri yapılması mümkündür. Çünkü ilk yörünge parçasının görece biçimde doğrultu ve hız değişikliği yapabileceği bir önceli bulunmamaktadır.

Değinilmesi gereken bir diğer nokta ise renk ve derinlik belleklerinde saklanan görselleştirme çıktılarının tamamının GİB belleğinden AİB belleğine transfer edilmesinin ve olası çözümleri bulmak adına doğrusal bir arama yapılmasının oldukça maliyetli olduğu gerçeğidir. Bunun yerine her bir yörünge parçası arama yinelemesi içerisinde, çözüm olarak kullanılacak pikseli belirlemek üzere GİB'nin görselleştirme iş hattını kullanan GLSL tabanlı iki GPGPU geçişi gerçekleştirilmektedir.

İlk GPGPU geçişi konum-zamansal görselleştirmenin hemen sonrasında gerçekleştirilir. İz düşüm resmini içeren renk belleği, doku resmi olarak ayarlanır ve onunla aynı boyutlara sahip bir diğer bellek alanı da çizim belleği olarak atanır. Ayarlanan doku resmi üzerindeki her bir doku hücresi için bir piksel işleyici tetiklenir. Bu piksel işleyiciler ilgili oldukları doku hücrelerini kontrol ederek doku hücresi koordinatlarının çizim belleğinde karşılık geldiği alana Şekil 5.8'de verilen algoritmaya göre bir KYM renk değeri yazar.

```

Input: Color (projection image bound as texture)
       Atomic counter variable white_solution_count
       x, y (texture coordinates of current fragment)
Output: Temp (Buffer bound as draw buffer)
white_solution_count = 0;
forall fragment shader instances in parallel do
  if Color[x, y].red == 0 then                                     /* is black */
    | Temp[x, y] = (-2, 0, 0);                                   /* black texel */
  end
  else if Color[x, y].red == 1 then                               /* is white */
    | mySpeed = Calculate resulting speed of solution using x and y;
    | if mySpeed ≥ minSpeed && mySpeed ≤ maxSpeed then
      | | if not searching for first segment then
      | | | deltaAngle = Calculate direction change;
      | | | deltaSpeed = Calculate speed change;
      | | | if deltaAngle le maxDeltaAngle && deltaSpeed le maxDeltaSpeed
      | | | then
      | | | | /* white texel and a possible solution */
      | | | | Temp[x, y] = (1, x, y);
      | | | | white_solution_count ++;
      | | | end
      | | | else
      | | | | /* white texel but not a solution */
      | | | | Temp[x, y] = (-1, 0, 0);
      | | | end
      | | end
      | | else
      | | | /* white texel and a possible solution */
      | | | Temp[x, y] = (1, x, y);
      | | | white_solution_count ++;
      | | end
      | end
      | else
      | | /* white texel but not a solution */
      | | Temp[x, y] = (-1, 0, 0);
      | end
    end
  end
end
end

```

Şekil 5.8 Çözüm olarak kullanılacak pikseli seçmek için gerçekleştirilen ilk geçişin algoritması

Doku hücresi siyah renkte ise, çıktı belleği üzerinde işlenen doku hücresine karşılık gelen alana sadece doku hücresinin renk durumunu ifade eden bir bilgi yazılır. Eğer doku hücresi beyaz renkte ise ve çözüm olarak kullanılması durumunda meydana getireceği yürüme hızı ön tanımlı yürüme hızı aralığında ise, önceki yörünge parçasına göreli doğrultu ve hız değişimleri kontrol edilir. Araması sürdürülen yörünge parçasının önceli varsa ve doku hücresinden türetilcek hız vektörünün bu öncele göre yapacağı doğrultu ve hız değişiklikleri *delta_angle* ve *delta_speed* eşik değerlerini aşmıyorsa, işlenmekte olan beyaz doku hücresi geçerli bir olası çözüm üretebiliyor demektir. Bu durumda doku hücresinin

koordinatlarının çizim belleği üzerinde karşılık geldiği alana, doku hücresinin koordinatları ile birlikte geçerli bir olası çözüm ürettiğini belirten bir bilgi yazılır. Buna ek olarak, olası çözümleri geçerli olan beyaz doku hücrelerinin sayısını tutan ve her yörünge parçası arama yinelemesinde sıfırlanan bir atomik sayacın değeri de bir artırılır. Beyaz bir doku hücresinin, kontrolü yapılan ön tanımlı sınırlamaların herhangi birine uymadığı tespit edildiğinde ise çizim belleğinde doku hücresinin ilgili alanına söz konusu bu beyaz doku hücresinin geçerli bir çözüm üretmediğini belirten bir bilgi yazılır.

İkinci GPGPU geçişi, ilk geçişin ardından atomik sayacın nihai değerine bağlı olarak ya tüm beyaz doku hücrelerinden ya da tüm siyah doku hücrelerinden türetilecek olası çözümlerin tamamını araştırır. Bu amaçla her bir doku hücresi için bir geometri işleyici tetiklenerek Şekil 5.9’da verilen GLSL tabanlı algoritmaya göre doku hücresinin meydana getireceği çözüm incelenir. İlk olarak atomik sayacın değeri kontrol edilir. Eğer atomik sayacın değeri sıfırdan büyük ise yolun güncel bitim noktasından karşı kenara kadar uzanan doğrusal, sabit hızlı ve çarpışmasız yörünge parçalarına karşılık gelen beyaz doku hücreleri mevcut demektir. Bu durumda sadece geçerli bir olası çözüm üreten beyaz doku hücrelerinin işlenmesine devam edilerek geçersiz çözüm üreten beyaz doku hücreleri ile diğer tüm siyah doku hücreleri yok sayılır ve bunlar için başka hiçbir işlem yapılmaz. Böylelikle GİB’nin iş hattında ilerleyen veri miktarı azaltılarak hesaplama yükü düşürülür. Bunun yanı sıra, ön tanımlı yürüme hızı aralığında hız değerine sahip olmakla birlikte öncel yörünge parçasına göreli doğrultu ve hız değişimleri belirlenen limitler dâhilinde olan çözümler üretilen de garanti edilmiş olur.

Tam tersine, eğer atomik sayaç içerisinde saklanan değer sıfıra eşit ise geçerli bir çözüm üreten hiçbir beyaz doku hücresi yok demektir. Buna rağmen, siyah doku hücrelerini kullanarak karşı kenara ulaşmayı başaramayan ara yörünge parçaları oluşturmak halen mümkündür. Bu durumda sadece siyah doku hücrelerinin işlenmesine devam edilirken tüm diğer beyaz doku hücreleri yok sayılır ve bunlar için başka hiçbir işlem yapılmaz.

Derinlik belleği içerisinde yer alan değerler, doğrudan doku hücreleri ile görüş hacminin yakın düzlemi arasındaki uzaklık olarak değil de doğrusal olmayan logaritmik benzeri bir fonksiyon ile bu uzaklığa bağlı olarak üretilen değerler biçiminde saklanmaktadır. Öncelikle, incelenen doku hücresinin gerçek derinlik değerini hesaplamak adına derinlik belleğinde kayıtlı değeri üzerinde gerekli doğrusallaştırma ve normalleştirme dönüşümleri uygulanır. Elde edilen bu gerçek derinlik değeri Eşitlik (5.2)’deki gibi hesaplanan ortalama yörünge parçası derinliği ile karşılaştırılır.

$$avg_segment_depth = \frac{d_{min}}{max_num_of_segments} \quad (5.2)$$

```

Input: Temp (output image of the previous pass)
          Depth (depth image bound as texture)
          Atomic counter variable white_solution_count
          x, y (texture coordinates of current primitive)
Output: Color (Buffer bound as draw buffer)
for all geometry shader instances in parallel do
    /* white texels exist as possible solution */
    if white_solution_count > 0 then
        /* white texel and a possible solution */
        if Temp[x, y].red == 1 then
            texel_pos = Calculate current texel's position using xorg and yorg;
            vertex.color = (-1, texel_pos.x, texel_pos.y);
            vertex.pos = (-1, -1, travel_timeinterval * scale_fac - texel_pos.y);
            emit(vertex);
        end
    end
    /* use black texels as possible solution */
    else if Temp[x, y].red == -2 then /* is black */
        myDepth = Calculate actual depth using depth value Depth[x, y];
        if myDepth < minDepth then
            Discard vertex; /* not a solution */
        end
        mySpeed = Calculate resulting speed of solution using x and y;
        if mySpeed < minSpeed || mySpeed > maxSpeed then
            Discard vertex; /* not a solution */
        end
        if not searching for first segment then
            deltaAngle = Calculate direction change;
            deltaSpeed = Calculate speed change;
            if deltaAngle > maxDeltaAngle || deltaSpeed > maxDeltaSpeed then
                Discard vertex; /* not a solution */
            end
        end
        texel_pos = Calculate current texel's position using xorg and yorg;
        vertex.color = (myDepth, texel_pos.x, texel_pos.y);
        vertex.pos = (-1, -1, myDepth);
        emit(vertex);
    end
end
end

```

Şekil 5.9 Çözüm olarak kullanılacak pikseli seçmek için gerçekleştirilen ikinci geçişin algoritması

Eğer siyah doku hücresinin gerçek derinlik değeri *avg_segment_depth* değerinden daha az ise en az *avg_segment_depth* derinliğine sahip en fazla *max_num_of_segments* adet yörünge parçası kullanarak karşı kenara ulaşmayı garanti etmek adına bu doku hücresi yok sayılır. Gerçek derinlik değerinin bu eşik değere eşit veya daha fazla olması durumunda ise siyah doku hücresinin oluşturacağı çözümün yürüme hızının, ilk geçişte olduğu gibi ön tanımlı

yürüme hızı aralığında olup olmadığı kontrol edilir ve bu aralık içerisinde olmaması durumunda siyah doku hücresi yine yok sayılır. Yürüme hızı aralığı koşulunun sağlanması durumunda ise siyah doku hücresi ile oluşacak çözümün doğrultu ve hız değerleri öncel yörünge parçasının doğrultu ve hız değerleri ile karşılıklı kıyaslanarak aralarındaki değişim farklarının *delta_angle* ve *delta_speed* parametreleri ile belirlenen kısıtlamalara uyup uymadığı kontrol edilir. Bu değişim farklarından herhangi birinin *delta_angle* ve *delta_speed* parametreleri ile belirtilen limit değerlerini aşması durumunda aynı şekilde siyah doku hücresi yok sayılır. Bu kontrol noktasından sonrasına kadar gelebilen siyah doku hücreleri ise ara yörünge parçaları için geçerli çözümlere karşılık gelir.

Nihayetinde, atomik sayıcin değerine göre sadece beyaz veya sadece siyah doku hücrelerinin ele alındığı, her iki durumda da yok sayılmadan bu noktaya kadar ulaşabilen ve geçerli çözümlere karşılık gelen her bir doku hücresinin bilgisi, çıktı belleğinin ilk alanına üst üste gelecek biçimde yazılır. Bu amaçla, yazılacak bu verileri renk bilgisi olarak taşıyan köşe ilkelleri, ortografik iz düşümde pozisyonları çakışık olacak biçimde yayınlanır. İkinci geçişin en başında GİB'nin derinlik testi [136] özelliği “daha fazla” (*greater*) fonksiyonunu kullanacak şekilde aktif edilir. Dolayısıyla bu köşe ilkellerinden, derinliği daha fazla olanın taşıdığı renk bilgisi, çıktı belleğindeki ilgili alana yazılarak çıktı belleği güncellenirken derinliği diğerlerine göre daha az olanlar yok sayılacaktır. Böylelikle ikinci geçiş tamamlandığında, çıktı belleğinin en başında yer alan ilk bellek alanında derinliği en fazla olan köşe ilkelinin taşıdığı bilgi yer alıyor olacaktır.

Buradan hareketle beyaz doku hücreleri için yayınlanan köşe ilkellerinin derinlik değerleri, doku hücresinin meydana getireceği çözümün hız değeri ile orantılı olacak biçimde ayarlanır. Bu sayede yeni birey oluşturmak için kullanılacak çözümün, doku hücrelerinin karşılık geldiği olası çözümlerin yürüme hızlarına göre seçilmesi sağlanır. Bu amaçla üç farklı seçim yöntemi kullanılabilir. İlk seçim yöntemi olarak en yüksek yürüme hızı değerine sahip olan çözümlerden birine karşılık gelen beyaz doku hücresinin seçilmesi sağlanır. Bu amaçla yayınlanacak köşe ilkellerinin derinliği, beyaz doku hücresinin dikey eksenindeki koordinatı (y_{coord}) kullanılarak belirlenir. Yüksek y_{coord} düşük yürüme hızına karşılık geleceği için köşe ilkellerinin derinliği, beyaz doku hücresinin dikey eksenindeki koordinatı, derinlik belleğini ortadan ikiye bölen yatak eksene göre simetrik olacak biçimde ($t_{int} * k - y_{coord}$) ayarlanır. Böylelikle geçişin sonunda en yüksek derinlikli köşe ilkelinin en yüksek yürüme hızlarından birini üreteceği garanti edilir. İkinci seçim yöntemi olarak en düşük yürüme hızı değerine sahip olan çözümlerden birine karşılık gelen beyaz doku hücresinin

seçilmesi sağlanır. Bu amaçla yayınlanacak köşe ilkellerinin derinliği, doğrudan y_{coord} olacak biçimde belirlenir. Yüksek y_{coord} düşük yürüme hızına karşılık geleceği için geçişin sonunda, en yüksek derinlikli köşe ilkelinin, en düşük yürüme hızlarından birini üreteceği garanti edilir. Son seçim yöntemi olarak geçerli çözüm üreten beyaz doku hücrelerinden birinin rastgele seçilmesi sağlanır. Bu amaçla geçerli çözümlere karşılık gelen beyaz doku hücreleri için yayınlanacak köşe ilkellerinin derinlikleri $[0, t_{int}]$ aralığında olacak biçimde rastgele atanır. Bu sayede geçişin sonunda rastgele seçilen derinlik değerlerinin en yükseğine ait bilgiler çıktı belleğinde bulunacağından olası çözümler arasında rastgele bir seçim yapılmış olur. ÇYPÇY yöntemi, varsayılan olarak birey oluşturma denemelerinin her bir yörünge parçası arama süreci boyunca, beyaz piksellerden elde edilen olası çözümler arasından en yüksek yürüme hızına sahip olan çözümlerden birini seçerek yörünge parçası oluşturacak şekilde ayarlanmıştır. Yine de bu seçim yöntemlerinin gerçek zamanlı benzetimlerde ortalama ve maksimum birey sayılarına etkilerini araştıran deneyler ve sonuçlarına “6.2.2 Çözüm Seçim Yöntemlerinin İrdelenmesi” bölümünde detaylıca yer verilmiştir.

Siyah doku hücreleri için yayınlanan köşe ilkellerinin derinlik değerleri ise daha önce açıklandığı gibi karşı kenara mümkün olan en yakın bitim noktasına sahip ara yörünge parçalarını oluşturmak üzere doku hücresinin gerçek derinlik değerine eşit olacak şekilde ayarlanır. Atomik sayacın değerine göre bu geçişin sonunda, çıktı belleğinin en başında yer alan ilk doku hücresinin konumuna, en yüksek derinlik değerine sahip olan; beyaz doku hücresinin (en yüksek, en düşük veya rastgele yürüme hızına karşılık gelen çözümlerden biri) ya da siyah doku hücresinin (bitim noktası karşı kenara en yakın ara yörünge parçasına karşılık gelen çözümlerden biri) bilgileri yazılır. Bu sayede tüm çıktı belleğinin GİB belleğinden AİB belleğine transfer edilmesi yerine sadece ilk doku hücresi konumundaki çıktı verisinin transfer edilmesi yeterli hale getirilerek veri transfer maliyeti azaltılır.

5.2. Bireylerin Hareket Ettirilmesi

Tez kapsamında geliştirilen ÇYPÇY yönteminde bireyler birden fazla doğrusal ve sabit hızlı yörünge parçasından oluşan çarpışmasız yollara sahip olabilmektedir. Bireylerin benzetim boyunca bu parçalı yollar üzerinde belirlenen sabit hız vektörleri ile yürümelerinin hiçbir çarpışmaya neden olmayacağı da garanti edilmiştir. Bu nedenle bireylerin benzetimde yer aldıkları süre boyunca çarpışma tespiti veya çarpışmadan kaçınılması için hiçbir işlem yapmasına gerek kalmamıştır.

Benzetim süresince bireylerin hareket ettirilmesi için her bir zaman adımında bireylerin pozisyonlarının basit bir biçimde güncellenmesi gerekmektedir. Ancak burada, geliştirilen

diğer iki gezinim yönteminden farklı olarak çarpışmasız yol boyunca kullanılan farklı yörünge parçalarından kaynaklanan farklı doğrultu ve yürüme hızları mevcuttur ve bireylerin pozisyonlarının güncellenmesi için o anda üzerinde buldukları yörünge parçasına karşılık gelen doğrultu ve hızın kullanılması gerekmektedir. Daha önce de belirtildiği üzere sabit bir doğrultu ve sabit bir yürüme hızı içeren her bir yörünge parçası aslında sabit bir hız vektörü ile de ifade edilebilmektedir. Dolayısıyla her bir zaman adımında bireylerin üzerinde buldukları yörünge parçasına karşılık gelen sabit hız vektörlerini kullanarak pozisyonlarının güncellenmesi işlemi gerçekleştirilebilir. Bu amaçla ilk önce bireylerin o an üzerinde buldukları yörünge parçasından kaynaklanan hız vektörleri belirlenir. Sonra bu hız vektörünün değeri TYPÇY yöntemlerinde tek yörünge parçasından oluşan çarpışmasız yollar üzerinde bireyleri hareket ettirmek için kullanılan Eşitlik (3.1)'deki formülde v hız vektörü yerine konularak bireylerin o anki hız vektörüne ve son zaman adımı içerisinde geçen süreye bağlı güncel pozisyonları hesaplanır. Ancak tahmin edilebileceği üzere, o an üzerinde buldukları yörünge parçasının bitim noktasına yakın olan bireylerin pozisyonlarının, bu işlem sonucunda ilgili yörünge parçasının bitim noktasına ulaşabilme ve hatta hareket doğrultusu üzerinde, yörünge parçasının bitim noktasından daha da öteye geçebilme ihtimalleri mevcuttur.

Bu durum, tek yörünge parçasından oluşan yollara sahip bireyler ya da çok yörünge parçasından oluşan yollarının son yörünge parçası üzerinde hareket etmekte olan bireyler için herhangi bir sorun teşkil etmeyecektir. Çünkü pozisyon güncellemesinin hemen sonrasında yine TYPÇY yöntemlerinde kullanılan ve Eşitlik (3.2)'de verilen formüle göre tüm bireylerin üzerinde buldukları yörünge parçalarının bitim noktalarına ulaşım ulaşmadığı kontrol edilir. Buna göre, örneklenen bu bireylerin yalnızca üzerinde buldukları yörünge parçasının bitim noktasına değil, aynı zamanda sahip oldukları çarpışmasız yolun bitiş noktasına da ulaştıkları ($\alpha = 90^\circ$) ve belki de daha ötesine geçtikleri ($\alpha > 90^\circ$) tespit edilerek benzetimden çıkarılmaları sağlanır. Bu sayede bireylerin, bitiş noktalarının üzerinde yer aldığı benzetim alanı kenarının ötesine geçerek benzetim alanının dışına çıkmaları engellenmiş olur.

Diğer yandan, birden çok yörünge parçasından oluşan yollara sahip olup henüz son yörünge parçası üzerinde olmayan bireyler de var olabilir. Bu gibi bireyler, üzerinde buldukları yörünge parçasının bitim noktasına ulaştığında yolları üzerindeki bir sonraki yörünge parçası ile devam etmek durumundadır. Bu bireyler için de pozisyon güncellemesinin hemen sonrasında, Eşitlik (3.2)'de verilen formüle göre üzerinde buldukları yörünge parçalarının

bitim noktalarına ulaşıp ulaşmadıkları kontrol edilir. Güncel pozisyonları bu bitim noktalarının tam üzerinde ($\alpha = 90^\circ$) olan bireyler aynı zamanda bir sonraki yörünge parçalarının da başlangıç noktaları üzerindedir. Bu bireylerin hız vektörü, sıradaki yörünge parçasından elde edilecek hız vektörü ile güncellenerek, benzetimin devamında bu yörünge parçasının başlangıç noktasından başlayıp onun üzerinde yürümeye devam etmeleri sağlanır. Güncel pozisyonları bu bitim noktalarının ötesine geçmiş ($\alpha > 90^\circ$) olan bireyler için ise öncelikle bu bitim noktalarından öteye ne kadar mesafe gittikleri (d_{diff}) hesaplanır. Sonrasında d_{diff} mesafesi bireyin o anki yürüme hızına bölünerek bu mesafeyi ne kadar sürede (t_{diff}) katettikleri bulunur. Bireylerin üzerinde buldukları yörünge parçalarının bitim noktalarına ulaştıktan sonra geçen t_{diff} süresi zarfında bir sonraki yörünge parçaları üzerinde bu yörünge parçasından kaynaklanan hız vektörü uyarınca yürümüş olmaları gerekir. Bunu sağlamak adına ilk önce bireylerin güncel pozisyonları, üzerinde buldukları yörünge parçalarının bitim noktaları ve aynı zamanda bunlarla aynı pozisyona karşılık gelen bir sonraki yörünge parçalarının başlangıç noktaları ile çakışık olacak biçimde güncellenir. Daha sonra bu bireylerin hız vektörü bir sonraki yörünge parçasından elde edilecek hız vektörü ile güncellenir. Son olarak, Eşitlik (3.1)'deki formüle göre bireylerin güncel pozisyonları yeniden hesaplanarak bitim noktalarına ulaştıktan sonra geçen t_{diff} süresi zarfında bu bitim noktasını takip eden yörünge parçasının doğrultu ve yürüme hızına göre ilerlemeleri ve aslında olmaları gereken konuma ulaşmaları sağlanır.

6. DENEYLER VE SONUÇLARI

Tez kapsamında geliştirilen üç farklı yönlendirmesiz gezinim yönteminin başarımını değerlendirmek adına çeşitli deneyler gerçekleştirilmiştir. Geliştirilen bu yöntemlerin öncelikli hedefi, mümkün olan en fazla sayıda bireyin benzetim boyunca herhangi bir çarpışma tespiti veya çarpışmadan kaçınma işlemi olmaksızın geziniminin sağlanmasıdır. Bunu yaparken de gezinim maliyetinin olabildiğince düşük tutularak alanda mevcut olan güncel birey tabanlı gezinim yöntemlerine üstünlük sağlanması hedeflenmiştir.

Bu hedeflere ne ölçüde ulaşıldığının incelenmesi adına, bu bölümde öncelikle TYPÇY kullanan gezinim yöntemlerinin birey oluşturma başarımları irdelenmiştir. Benzer biçimde, daha sonra ÇYPÇY yönteminin birey oluşturma başarımı değerlendirilmiştir. Bunu takiben her üç gezinim yönteminin birey oluşturma ve gezinim maliyetleri hem birbirleriyle hem de alanda mevcut güncel bir diğer yöntem ile kıyaslanmıştır. Devamında TYPÇY ve ÇYPÇY yöntemlerinin farklı şekil ve sayılarda sabit engeller içeren farklı şekil ve büyüklükteki benzetim alanları üzerindeki birey oluşturma başarımları incelenmiştir. Son olarak, geliştirilen yönlendirmesiz gezinim yöntemlerindeki bireylerin yörünge kalitesi, alanda mevcut güncel birey tabanlı gezinim yöntemlerinden biri ile karşılaştırılmıştır.

Bu bölümde yapılan tüm testlerde:

- 3.4 GHz saat frekansına sahip Intel i7 2600K model bir AİB kullanılmıştır.
- 8 GB büyüklüğünde ana bellek kullanılmıştır.
- 1.5 GB sistem belleğine sahip Nvidia GTX580 model bir GİB kullanılmıştır.
- Tüm bireylerin yürüme hızlarının $[1.0, 2.0]$ m/s sürekli aralığı içerisinde değerlere sahip olması sağlanmıştır.
- AİB tabanlı TYPÇY yönteminde konum-zamansal dizinleme veri yapısı, bir kenarının uzunluğu 4 m olan kare biçiminde hücrelerden oluşan tekdüze bir ızgara biçimindedir.
- Daha önce de bahsedildiği üzere çözüm uzayının çözünürlüğünü ayarlamak için kullanılan k parametresinin değeri $k = 1$ olacak biçimde atanarak GİB üzerinde gerçekleştirilen çarpışmasız çözüm arama sürecinde bir doku hücresinin çözüm uzayı içerisinde 1 m^2 alana karşılık gelmesi sağlanmıştır.

6.1. TYPÇY Yöntemlerinin Değerlendirilmesi

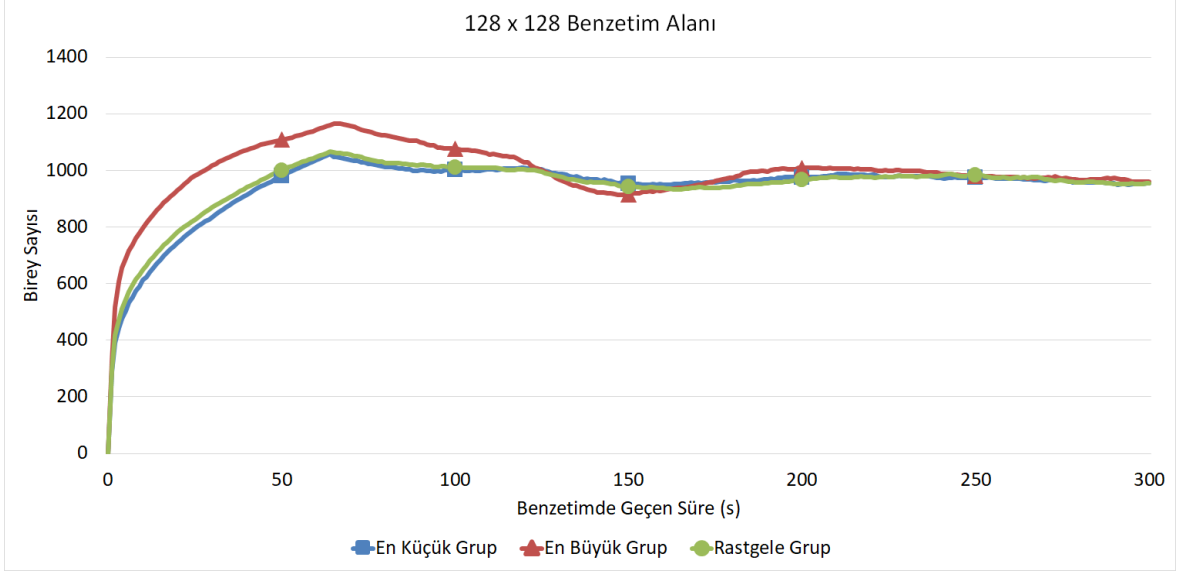
6.1.1. Grup Merkezi Seçim Yöntemlerinin İrdelenmesi

TYPÇY'ler üzerinde bireyleri herhangi bir manevra yapmaksızın hareket ettiren GİB tabanlı TYPÇY yönteminin detayları anlatılırken söz edildiği üzere her yeni birey oluşturma sürecinde, yakalanan iz düşüm resmi üzerindeki bitişik beyaz piksel gruplarının merkez noktaları hesaplanarak bu noktalardan yalnız biri çözüm olarak kullanılmıştır. Çözüm olarak kullanılacak merkez noktasının ait olduğu grubun seçilmesi için ise üç farklı yöntem önerilmiştir. Bunlardan ilki en fazla sayıda komşu beyaz pikseli içeren en büyük grup yöntemi, ikincisi en az sayıda komşu beyaz pikseli barındıran en küçük grup yöntemi, üçüncüsü ise tüm beyaz piksel grupları arasından birinin rastgele seçildiği rastgele grup yöntemidir.

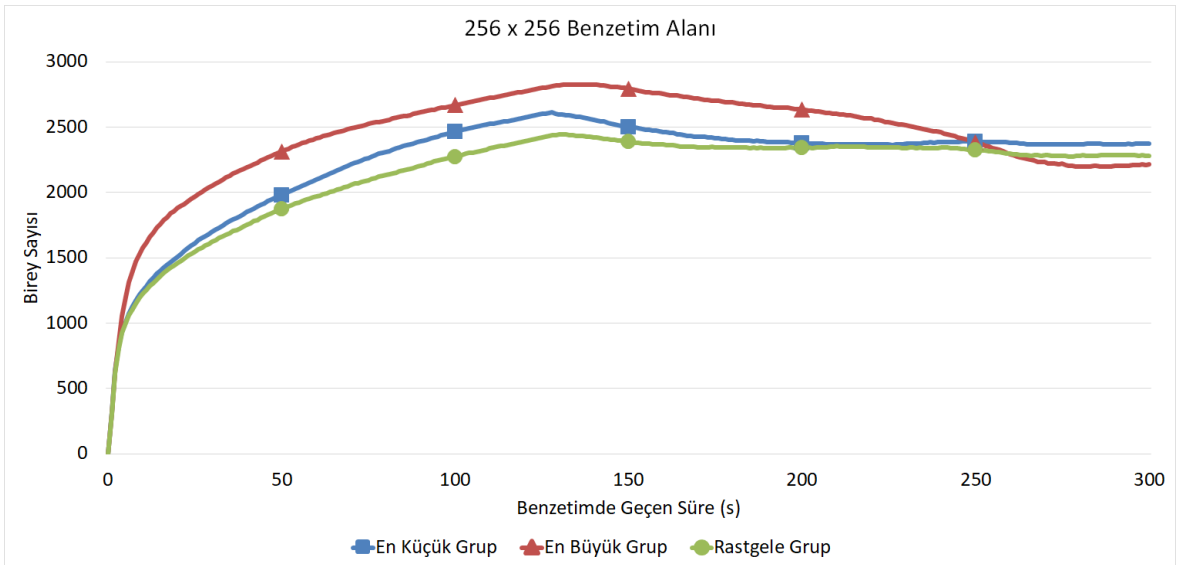
Bu seçim yöntemlerinin birey oluşturma başarımını incelemek adına bir kenarının uzunluğu $128\ m$ ve $256\ m$ olan kare biçiminde iki farklı benzetim alanı kullanılmıştır. Her bir grup merkezi seçim yöntemi için bu benzetim alanları üzerinde 5 dakika sürecek beş farklı benzetim çalıştırılmıştır. Tüm bu benzetimler boş bir benzetim alanı ile başlamakta ve benzetim boyunca bu benzetim alanını olabildiğince fazla birey ile doldurmaya çalışmaktadır. Benzetimlerde yer alan tüm bireylerin yarıçapları, $0.5\ m$ olacak biçimde ayarlanmıştır. Bu arada tüm benzetimler, her bir benzetim zaman adımında yalnızca bir kez birey oluşturma denemesi yapacak biçimde ayarlanmıştır. Benzetimlerin her bir zaman adımında benzetimde yer alan birey sayıları ölçülerek her bir saniye başına bu bir saniye içerisinde kalan tüm ölçümlerin ortalaması alınmıştır. Daha sonra aynı benzetim alanı üzerinde aynı seçim yöntemini kullanacak biçimde çalıştırılan beş farklı benzetim boyunca saniye bazında elde edilen bu ortalama birey sayılarının karşılıklı her bir saniye için ortalaması alınarak yöntemlerin genel davranışlarını yansıtacak nihai sonuçlar elde edilmiştir.

$128\ m$ ve $256\ m$ kenarlara sahip kare benzetim alanları üzerinde elde edilen ve benzetim boyunca saniye bazında ortalama birey sayılarını gösteren sonuçlar sırasıyla Şekil 6.1 ve Şekil 6.2'de verilmiştir. Elde edilen bu sonuçlar, her birey oluşturma denemesinde en büyük beyaz piksel grubuna ait küme merkezinin seçilerek çözüm olarak kullanılmasının her iki benzetim alanını bireyler ile doldurma konusunda alternatif diğer iki seçim yönteminden daha hızlı olduğunu göstermektedir. Buradan hareketle benzetimin başlarında en büyük grup yönteminin birey oluşturma başarısının rakiplerine kıyasla daha yüksek olduğunu söylemek mümkündür. Buna ek olarak, en büyük grup merkezini seçen yöntemin diğer iki yöntemle

kıyasla her iki benzetim alanı üzerinde de en yüksek maksimum birey sayılarına ulaşabildiği görülmektedir. Bu durum ise, en büyük grup yönteminin yoğun kalabalıklar içerisinde birey oluşturma başarısının diğer yöntemlere göre daha yüksek olmasıyla açıklanmaktadır. Yine de benzetim alanı büyüklüğünden bağımsız olarak bu üç farklı yöntemin, nihayetinde yaklaşık birey sayısı değerlerine yakınsadığı tespit edilmiştir.



Şekil 6.1 Grup merkezi seçim yöntemlerinin 128 m kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları



Şekil 6.2 Grup merkezi seçim yöntemlerinin 256 m kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları

Ayrıca benzetimlerden elde edilen birey sayısı verileri üzerinde birtakım istatistiksel analizler yapılmıştır. Bu amaçla gerçekleştirilen her bir benzetim boyunca ulaşılan ortalama ve maksimum birey sayıları hesaplanmıştır. Daha sonra bu ortalama ve maksimum birey sayılarının bağımlı değişkenler olarak kullanıldığı ve grup merkezi seçim yöntemi ile benzetim alanı büyüklüğünün bağımsız değişkenler olarak ele alındığı iki yönlü bir “çoklu varyans analizi” (*Multivariate Analysis of Variance - MANOVA*) gerçekleştirilmiştir. “Çoklu karşılaştırma” (*Post-hoc*) analizleri Tukey’in “dürüst anlamlı fark” (*Honest Significant Difference - HSD*) testi kullanılarak gerçekleştirilmiş ve %95 güven seviyesinde istatistiksel olarak anlamlı tüm bulgular rapor edilmiştir. Gerçekleştirilen analizler neticesinde, benzetim alanı büyüklüğünün, ortalama birey sayısı ($F(1, 26) = 3896.443, p \approx 0$) ve maksimum birey sayısı ($F(1, 26) = 3307.737, p \approx 0$) üzerinde ana etkiye sahip olduğu tespit edilmiştir. Buna göre, benzetim alanı büyüklüğü arttıkça hem ortalama birey sayısının hem de maksimum birey sayısının arttığını söylemek mümkündür. Diğer yandan, kullanılan grup merkezi seçim yönteminin ortalama birey sayısı ($F(2, 26) = 21.456, p \approx 0$) ve maksimum birey sayısı ($F(2, 26) = 28.631, p \approx 0$) üzerinde ana etkiye sahip olduğu gözlemlenmiştir. Çoklu karşılaştırma testlerine göre en büyük grup yöntemi, hem ortalama birey sayısında hem de maksimum birey sayısında en küçük grup ve rastgele grup yöntemlerinden daha fazla bireye ulaşabilmektedir (tüm durumlar için $p \approx 0$). Buna rağmen, en küçük grup yöntemi ile rastgele grup yöntemi arasında hem ortalama birey sayısı ($p < 0.203$) hem de maksimum birey sayısı ($p < 0.055$) göz önünde bulundurulduğunda istatistiksel açıdan anlamlı bir fark olmadığı görülmüştür. Ayrıca benzetim alanı büyüklüğü ve grup merkezi seçim yöntemi bağımsız değişkenlerinin etki büyüklükleri (η^2) Çizelge 6.1’de verildiği gibidir.

Çizelge 6.1 Grup merkezi seçim yöntemi ve benzetim alanı büyüklüğünün etki büyüklükleri

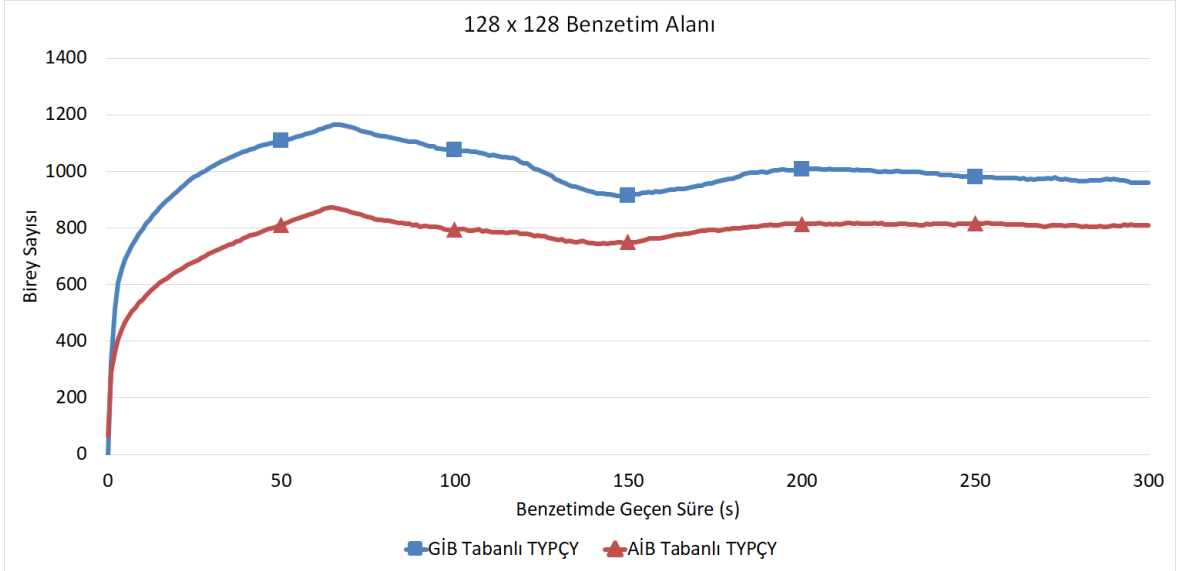
Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Benzetim alanı büyüklüğü	1	3896.443	0.000	0.983
	Grup merkezi seçim yöntemi	2	21.456	0.000	0.011
	Hata	26			
Maksimum birey sayısı	Benzetim alanı büyüklüğü	1	3307.737	0.000	0.975
	Grup merkezi seçim yöntemi	2	28.631	0.000	0.017
	Hata	26			

6.1.2. Birey Oluşturma Başarımının Karşılaştırılması

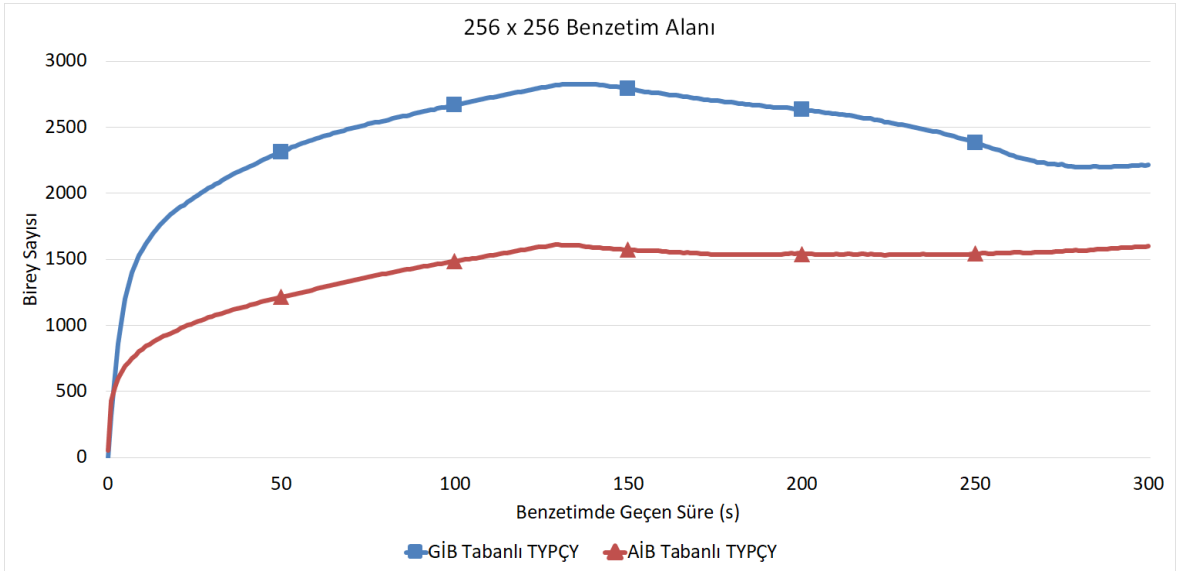
Bu deney kapsamında, geliştirilen AİB tabanlı TYPÇY yöntemi ile GİB tabanlı TYPÇY yöntemi, bir önceki bölümde kullanılan ortalama birey sayısı ölçüsü üzerinden karşılaştırılmıştır. Karşılaştırılan her iki gezinim yöntemi de TYPÇY'ler oluşturmakta ve bu yolları yeni oluşturulacak bireylere tahsis etmektedir. Hatırlanacağı üzere AİB tabanlı TYPÇY yöntemi, çarpışmasız yolları belirlemek üzere tamamı AİB üzerinde çalışan ve başlangıç pozisyonu, bitiş pozisyonu ve yürüme hızı parametrelerinin tamamını rastgele seçerek karşılık gelen yolun var olan bireylerle çarpışma içerip içermediğini kontrol eden bir birey oluşturma süreci kullanmaktadır. Söz konusu bu yöntemin ona alternatif olarak geliştirilen GİB tabanlı TYPÇY yöntemi ile karşılaştırılması sürecinde, GİB tabanlı TYPÇY yönteminin olası çözümlerden birini seçmesini sağlamak üzere önceki bölümdeki sonuçlara dayanılarak en başarılı olduğu görülen en büyük grup merkezi yönteminin kullanılmasına karar verilmiştir. Yine önceki bölümde tanımlanan farklı büyüklüklere sahip iki benzetim alanı üzerinde, her biri 5 dakika süren ve boş benzetim alanını doğrusal yörüngeler üzerinde yürüten bireyler ile doldurmaya çalışan beş farklı benzetim çalıştırılmıştır. Her iki gezinim yöntemi ile yapılan benzetimlerde de birey oluşturma işlemleri için her bir benzetim zaman adımında sabit 2.5 ms süre ayrılmış ve bu süre zarfında başarılı veya başarısız olduğuna bakmaksızın benzetimlerin art arda mümkün olduğunca çok birey oluşturma denemesi yapması sağlanmıştır. Benzetimlerde yer alan tüm bireylerin yarıçapları, 0.5 m olacak biçimde ayarlanmıştır. İki farklı gezinim yöntemi için çalıştırılan benzetimlerin kendi içinde ortalaması alınarak benzetimin her bir saniyesindeki ortalama birey sayıları hesaplanmıştır.

128 m ve 256 m kenarlara sahip kare benzetim alanları üzerinde elde edilen ve gezinim yöntemlerinin benzetim boyunca saniye bazında ortalama birey sayılarını gösteren sonuçlar sırasıyla Şekil 6.3 ve Şekil 6.4'te verilmiştir. Bu sonuçlara göre, GİB tabanlı TYPÇY yöntemi her iki benzetim alanını da AİB tabanlı TYPÇY yöntemine göre daha hızlı doldurmakla kalmayıp bu alanlar üzerinde sırasıyla %33 ve %75 daha yüksek maksimum birey sayılarına ulaşılmasını sağlamıştır. Diğer bir deyişle, GİB tabanlı TYPÇY yöntemi birim zamanda daha fazla birey oluşturabilme yeteneğinin sonucu olarak daha yoğun yönlendirmesiz kalabalıklar meydana getirmeyi başarmış ve bu bakımdan karşılaştırıldığı AİB tabanlı TYPÇY yönteminden üstün gelme başarısını göstermiştir. Bu sonucun temel nedeni, AİB tabanlı TYPÇY yönteminin çarpışmasız yolları belirlemek üzere yalnızca rastgele üretilmiş parametre değerlerini kullanması ve bu rastgeleliğin her bir benzetim zaman adımında birey oluşturma işlemleri için ayrılan süre boyunca çok sayıda başarısız

denemeyle sonuçlanmıştır. Diğer taraftan, GİB tabanlı TYPÇY yöntemi, GİB'nin görselleştirme iş hattından faydalanarak seçilen başlangıç pozisyonundan karşı kenara kadar ulaşan tüm olası çarpışmasız yol çözümlerinin tek seferde tespit edilmesine olanak sağlayan çok daha “belirleyici” (*deterministic*) bir yaklaşım kullanmaktadır.



Şekil 6.3 Gezinim yöntemlerinin 128 m kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları



Şekil 6.4 Gezinim yöntemlerinin 256 m kenara sahip kare benzetim alanı üzerinde elde edilen sonuçları

Deney kapsamında elde edilen birey sayısı verileri üzerinde çeşitli istatistiksel analizler gerçekleştirilmiştir. Bu doğrultuda her bir benzetim için ortalama ve maksimum birey sayıları hesaplanmıştır. Benzetim verilerinden üretilen ortalama ve maksimum birey sayılarını bağımlı değişken olarak kullanırken gezinim yöntemi ile benzetim alanı büyüklüğünü bağımsız değişkenler olarak ele alan %95 güven seviyesinde iki yönlü bir MANOVA gerçekleştirilmiştir. Analiz sonuçları incelendiğinde benzetim alanı büyüklüğünün ortalama birey sayısı ($F(1, 17) = 117.538, p \approx 0$) ve maksimum birey sayısı ($F(1, 17) = 114.442, p \approx 0$) üzerinde ana etkiye sahip olduğu gözlemlenmiştir. Bu durumda benzetim alanı büyüklüğü arttıkça hem ortalama birey sayısının hem de maksimum birey sayısının arttığını söylemek mümkündür. Ayrıca kullanılan gezinim yönteminin de ortalama birey sayısı ($F(1, 17) = 40.720, p \approx 0$) ve maksimum birey sayısı ($F(1, 17) = 45.119, p \approx 0$) üzerinde ana etkiye sahip olduğu tespit edilmiştir. Buna göre, GİB tabanlı TYPÇY yönteminin, rakip olduğu AİB tabanlı TYPÇY yöntemine kıyasla aynı büyüklükte benzetim alanları üzerinde hem ortalama birey sayısını hem de maksimum birey sayısını istatistiksel açıdan anlamlı bir oranda artırdığı görülmüştür. Son olarak, benzetim alanı büyüklüğü ve gezinim yöntemi bağımsız değişkenlerinin etki büyüklükleri Çizelge 6.2’de verildiği gibidir.

Çizelge 6.2 Gezinim yöntemi ve benzetim alanı büyüklüğünün etki büyüklükleri

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Benzetim alanı büyüklüğü	1	117.538	0.000	0.671
	Gezinim yöntemi	1	40.720	0.000	0.232
	Hata	17			
Maksimum birey sayısı	Benzetim alanı büyüklüğü	1	114.442	0.000	0.648
	Gezinim yöntemi	1	45.119	0.000	0.256
	Hata	17			

6.2. ÇYPÇY Yönteminin Değerlendirilmesi

6.2.1. Birey Oluşturma Başarımının Karşılaştırılması

Tez kapsamında geliştirilen ÇYPÇY yönteminin birey sayısı bakımından avantajlarını göstermek adına, AİB tabanlı TYPÇY metodunun yanı sıra GİB tabanlı TYPÇY yöntemi ile de kıyaslanması öngörülmüştür. Önceki bölümde verilen sonuçlarda da görüldüğü üzere, AİB tabanlı TYPÇY metoduna benzer biçimde TYPÇY’ler kullanan GİB tabanlı TYPÇY yöntemi dahi hâlihazırda bu metottan daha yüksek birey sayılarına ulaşabilmektedir. Burada incelenecek olan ÇYPÇY yönteminin ÇYPÇY’ler kullanarak olası çözüm sayısını ve

dolayısıyla birey sayısını GİB tabanlı TYPÇY yöntemine göre daha da artıracığı öngörülmektedir. GİB tabanlı TYPÇY yönteminden daha yüksek birey sayılarına ulaşmak, aynı zamanda bu AİB tabanlı TYPÇY yönteminden de daha yüksek birey oluşturma başarımı anlamına gelmektedir.

Burada karşılaştırılan tüm gezinim yöntemleri, benzetimin her bir zaman adımında yeni çarpışmasız yollar bularak yeni bireyler oluşturmak için çabalamaktadır. Ancak benzetimde yer alan birey sayısının artmasıyla birlikte yeni bir çarpışmasız yol bulma olasılığı da giderek azalmakta ve bu durum yeni birey oluşturmayı daha da zor hale getirmektedir. Ayrıca aynı benzetim zaman adımı içerisinde yeni bir birey oluşturuncaya dek beklemek de pratikte mümkün değildir. Çünkü bu durum, benzetimlerin gerçek zamanda çalışma hedefine aykırı olacaktır. Bu nedenle tüm bu yöntemlerde, her bir zaman adımı içerisinde birey oluşturma işlemleri için sınırlı ve sabit bir süre belirlenir ve benzetimlerin bu süre içerisinde başarılı veya başarısız olduğuna bakılmaksızın arka arkaya birey oluşturma denemeleri yapması sağlanarak mümkün olduğunca fazla sayıda birey oluşturma hedeflenir.

Benzetim zaman adımları içerisinde birey oluşturma işlemleri için ayrılan süre arttıkça her bir zaman adımında daha çok birey oluşturma denemesi yapılabilir. Bu gibi bir durumda başarıyla sonuçlanan birey oluşturma denemelerinin sayısındaki artışa bağlı olarak benzetimde yer alan bireylerin sayısının da artacağı yönünde bir çıkarım yapmak mümkündür. Bu çıkarımı test etmek üzere birey oluşturma için ayrılan sürenin farklı değerleri ele alınmalıdır. Bu amaçla birey oluşturma başarımının karşılaştırıldığı test sürecinde, birey oluşturma için ayrılan süre olarak 5, 10 ve 25 ms olmak üzere üç farklı değer kullanılmıştır. Bu değerler, sonraki bölümlerde (6.3.1 Birey Oluşturma Maliyetlerinin Karşılaştırılması) detayları verilecek olan ve geliştirilen yöntemlerin maliyet incelemesinin yapıldığı deneylerde ölçülen bir birey oluşturma için gereken ortalama süre metriğinin değerinden yola çıkılarak her bir zaman adımında birden fazla sayıda birey oluşturma denemesi yapılmasını sağlayacak biçimde seçilmiştir.

Hatırlanacağı üzere ÇYPÇY yöntemi, GİB tabanlı TYPÇY yöntemine göre fazladan bazı parametreler kullanmaktadır. Bunlardan çarpışmasız bir yolun sahip olabileceği yörünge parçası sayısının üst limitini belirleyen *max_num_of_segments* parametresi için 4, 8 ve 16 olmak üzere üç farklı değer kullanılmıştır. Buna göre, benzetimlerde kullanılacak tüm yollar, *max_num_of_segments* parametresinin seçilen değerine bağlı olarak en fazla 4, 8 veya 16 yörünge parçasına sahip olabilecektir. Ardışık yörünge parçası aramalarında, ara yörünge parçalarının bitim noktalarının geri çekilme payını belirleyen *margin_coef* parametresi için

2, 4 ve 6 olmak üzere üç farklı değer kullanılmıştır. Buna göre, ardışık yörünge parçası aramalarında, ara yörünge parçalarının bitim noktaları *margin_coef* parametresinin seçilen değerine bağlı olarak $2 * r$, $4 * r$ veya $6 * r$ kadar geri çekilecektir. Ardışık yörünge parçalarının doğrultuları arasındaki kırılma açısının üst limitini belirleyen *delta_angle* parametresi için 10° , 20° ve 30° olmak üzere üç farklı değer kullanılmıştır. Buna göre, *delta_angle* parametresinin seçilen değerine bağlı olarak ardışık yörünge parçaları arasında, bireyler en fazla 10° , 20° veya 30° kadar doğrultu değişikliği yapabilecektir. Ardışık yörünge parçalarının yürüme hızları arasındaki farkın üst limitini belirleyen *delta_speed* parametresi için 0.1, 0.3 ve 0.5 m/s olmak üzere üç farklı değer kullanılmıştır. Buna göre, *delta_speed* parametresinin seçilen değerine bağlı olarak ardışık yörünge parçaları arasında bireyler, yürüme hızlarını en fazla 0.1, 0.3 ve 0.5 m/s kadar artırabilecek veya azaltabilecektir.

Kullanılan bu parametrelerin herhangi birinin artırılmasının, artan işlem maliyeti pahasına birey sayısını da artıracığı yönünde bir hipotez kurulmuştur. Bu nedenle bu parametrelerin deneylerde kullanılacağı belirtilen değerleri, ÇYPÇY yönteminin potansiyel avantajlarını minimum maliyetle açığa çıkarmak üzere olabildiğince düşük seviyede seçilmiştir. Bu parametrelerin her bir kombinasyonu için, bir kenarının uzunluğu 128 m olan kare biçiminde bir benzetim alanı üzerinde 5 dakika süren benzetimler çalıştırılmıştır. Tüm benzetimler boş bir benzetim alanı ile başlayıp bu alanı mümkün olduğunca fazla sayıda bireyle doldurmak için çalışmaktadır. Benzetimlerin her bir saniyesinde, benzetimde yer alan bireylerin sayısı ölçülerek benzetim sonunda ortalama ve maksimum birey sayıları hesaplanmıştır. Benzer biçimde, sadece birey oluşturma için ayrılan süre parametresine sahip AİB tabanlı TYPÇY metodunun yanı sıra GİB tabanlı TYPÇY yöntemi de kullanılarak bu parametrenin her bir farklı değeri için aynı şekilde bir benzetim çalıştırılmış ve aynı ortalama ve maksimum birey sayısı metriklerinin değerleri hesaplanmıştır. Tüm bunlara ek olarak, ÇYPÇY yöntemini diğer yöntemler ile doğrudan kıyaslayabilmek adına sadece TYPÇY'lere sahip bireyler oluşturmaya izin verilen özel benzetimler gerçekleştirilmiştir. Birey oluşturma için ayrılan süre parametresinin her bir farklı değeri için bu şekilde özel bir benzetim çalıştırılarak aynı ortalama ve maksimum birey sayısı metriklerinin değerleri hesaplanmıştır.

Parametrelerin farklı kombinasyonlarına karşılık gelen benzetimlerin her birini biricik isimlerle adlandırmak üzere, ilgili kombinasyonu oluşturan parametrelerin değerleri *max_num_of_segments*, *margin_coef*, *delta_angle*, *delta_speed* sırasında ve aralarında alt tire “_” işareti bulunacak şekilde birleştirilerek biricik isimler oluşturulmuştur. Örneğin; *4_2_10_0.1* isimli benzetimde, herhangi bir çarpışmasız yolun sahip olabileceği maksimum

yörünge parçası sayısı 4, yörünge parçası bitim noktalarının geri çekilme payı $2 * r$, ardışık yörünge parçaları arasında izin verilen maksimum doğrultu değişikliği açısı 10° ve ardışık yörünge parçaları arasında izin verilen maksimum yürüme hızı değişikliği $0.1 m/s$ olacaktır. GİB tabanlı TYPÇY yöntemiyle gerçekleştirilen benzetimleri ifade etmek adına aynı formata sahip “0_0_0_0” özel ismi kullanılmıştır. Ayrıca ÇYPÇY yönteminin sadece TYPÇY’lere sahip bireyler oluşturmasına izin verildiği benzetimler de yine aynı formata sahip “1_0_0_0” özel ismi ile adlandırılmıştır.

Tüm bu deneyler, birey yarıçapının iki farklı değeri için iki farklı grup olarak gerçekleştirilmiştir. İlk gruptaki deneylerde, alanda var olan bazı çalışmalara [137], [138] benzer biçimde birey yarıçapı olarak $0.5 m$ değeri kullanılmıştır. Bu sayede tüm bireylerin, yürüme animasyonları dâhilinde yaptıkları adım atma ve kol sallama hareketleri esnasında, hareketli uzuvlarının da diğer hiçbir birey ile çarpışmamasını garanti etmek adına geniş ve güvenli dairesel sınırlayıcı alanlara sahip olmaları sağlanmıştır. İkinci grup deneylerde ise birey yarıçapı olarak $0.3 m$ değeri kullanılmıştır. Bu sayede bireylerin, Fruin [139] tarafından yaklaşık $58 cm$ olarak belirtilen ortalama insan vücudunun yanal genişliğine yakın çap değeri ($2 * r = 0.6 m$) ile ifade edilen dairesel sınırlayıcı alanlara sahip olmaları sağlanmıştır. Bu çap değeri ($0.6 m$) aynı zamanda Helbing ve arkadaşları [23] tarafından geliştirilen yöntemde, bireylerin sahip oldukları sınırlayıcı alanların belirlenmesi için kullanılan futbol taraftarlarının omuz genişliği dağılımının ortalama değerine karşılık gelmektedir.

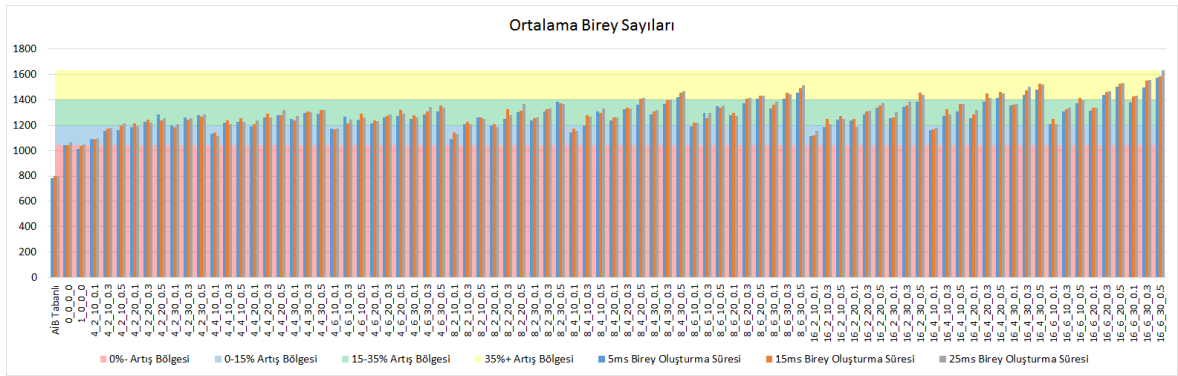
6.2.1.1. Birey Yarıçapının 0.5 m Değeri için Sonuçlar

Her bir benzetimin ortalama ve maksimum birey sayılarını gösteren grafikler, sırasıyla Şekil 6.5 ve Şekil 6.9’da verilmiştir. Ayrıca sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, ortalama birey sayısı için sırasıyla Şekil 6.6, Şekil 6.7 ve Şekil 6.8’de, maksimum birey sayısı için sırasıyla Şekil 6.10, Şekil 6.11 ve Şekil 6.12’de verilmiştir. Bu sonuçlar göz önünde bulundurulduğunda, GİB tabanlı TYPÇY yönteminin, AİB tabanlı TYPÇY metoduna kıyasla birey oluşturma için ayrılan süre parametresinin değerine bağlı olarak ortalama birey sayısında %30-34 ve maksimum birey sayısında %41-48 daha yüksek değerlere ulaşabildiği görülmüştür. Benzer biçimde, ÇYPÇY yönteminin yalnızca TYPÇY’ler kullanacak şekilde gerçekleştirilen benzetimlerinin, AİB tabanlı TYPÇY metoduna göre, ortalama birey sayısında %30-31 ve maksimum birey sayısında %43-47 daha yüksek değerlere ulaşabildiği belirlenmiştir. GİB tabanlı TYPÇY ve ÇYPÇY yöntemleriyle yalnızca TYPÇY’ler kullanacak şekilde gerçekleştirilen benzetimlerde, hem

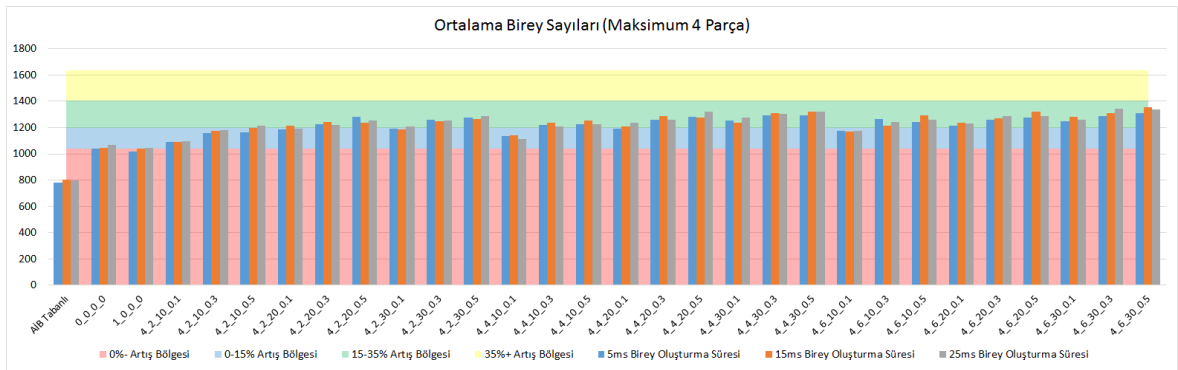
ortalama hem de maksimum birey sayıları arasında GİB tabanlı TYPÇY yöntemi lehine %1-2 gibi küçük farklar bulunmaktadır. Söz konusu bu küçük farklılıkların benzetimler boyunca oluşturulan bireylerin başlangıç pozisyonunu belirlemede kullanılan rastgelelikten kaynaklandığı düşünülmektedir. Yine de GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin yalnızca TYPÇY'ler kullanacak şekilde, birey oluşturma için ayrılan süre parametresinin yukarıda belirtilen farklı değerleri için tekrarlanan benzetimlerinden elde edilen ortalama ve maksimum birey sayıları üzerinde %95 güven seviyesinde iki yönlü bir MANOVA gerçekleştirilmiştir. Ortalama ve maksimum birey sayıları bağımlı değişkenler olarak kullanılırken gezinim yöntemi ve birey oluşturma için ayrılan süre parametreleri bağımsız değişkenler olarak ele alınmıştır. Çoklu karşılaştırma analizleri ise Tukey HSD testi kullanılarak gerçekleştirilmiştir. Birey sayıları üzerinde gerçekleştirilen bu istatistiksel analizler sonucunda, gerek ortalama ($F(1, 26) = 30.216, p \approx 0$) gerekse maksimum birey sayıları ($F(1, 26) = 25.751, p \approx 0$) arasındaki bu farkların istatistiksel açıdan anlamlı olduğu tespit edilmiştir (Çizelge 6.3). Bu küçük farklılıkların, GİB tabanlı TYPÇY yönteminin, yalnızca TYPÇY'ler kullanan ÇYPCY yöntemine göre nispeten daha düşük birey oluşturma maliyetine sahip olması nedeniyle (6.3.1 Birey Oluşturma Maliyetlerinin Karşılaştırılması) her bir zaman adımında birey oluşturma için ayrılan süre içerisinde, daha fazla birey oluşturma denemesi yapabilmemesinden kaynaklandığı düşünülmektedir. Buna karşın, sadece TYPÇY'ler kullanan bu gezinim yöntemleriyle gerçekleştirilen benzetimlerde, her bir zaman adımı içerisinde birey oluşturma için ayrılan sürenin artırılmasının, beklenilen aksine ortalama birey sayısı ($F(2, 26) = 1.942, p < 0.164$) ya da maksimum birey sayısı ($F(2, 26) = 1.988, p < 0.157$) üzerinde herhangi bir etkisi olmadığını söylemek mümkündür (Çizelge 6.3). Bu durum, söz konusu gezinim yöntemleriyle yukarıda belirtilen yapılandırmaya sahip olacak biçimde gerçekleştirilen gerçek zamanlı benzetimlerin, birey oluşturma için ayrılan süre parametresinin en düşük değeri olan 5 ms süresinin kullanıldığı durumlarda dahi maksimum birey doygunluğuna ulaştığı gerçeğini ortaya çıkarmıştır. Burada olduğu gibi kalabalık yoğunluğu için bir üst sınır teşkil eden doygunluk noktasına ulaşılması durumunda, birey oluşturma için ayrılan sürenin ve buna bağlı olarak gerçekleştirilen birey oluşturma denemesi sayısının artırılmasına rağmen herhangi bir birey oluşturmak deyim yerindeyse imkânsız hale gelmektedir.

Çizelge 6.3 GİB tabanlı TYPÇY ve yalnızca TYPÇY'ler kullanan ÇYPÇY yöntemleri ile birey oluşturma için ayrılan süre parametresinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)

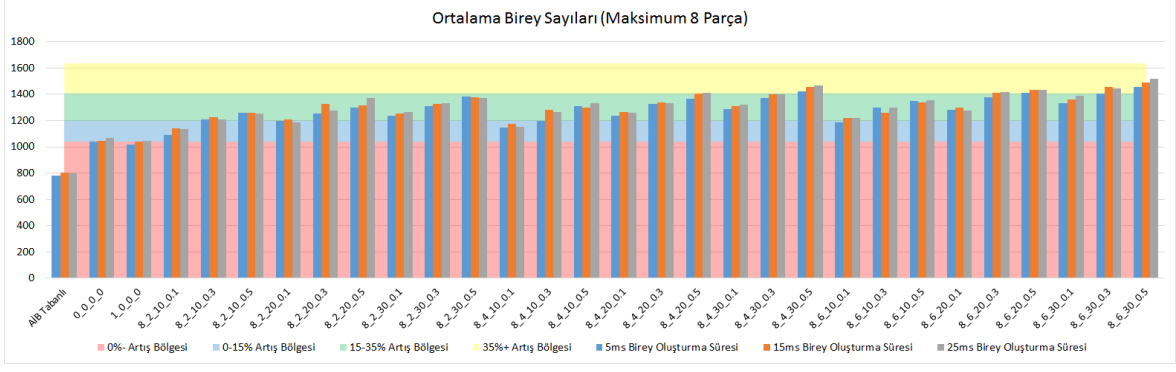
Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Gezinim yöntemi	1	30.216	0.000	0.503
	Birey oluşturma için ayrılan süre	2	1.942	0.164	0.065
	Hata	26			
Maksimum birey sayısı	Gezinim yöntemi	1	25.751	0.000	0.462
	Birey oluşturma için ayrılan süre	2	1.988	0.157	0.071
	Hata	26			



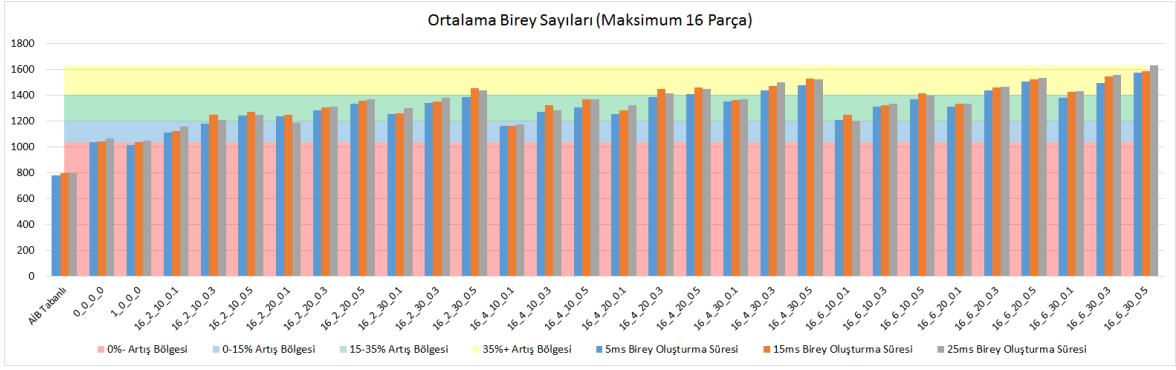
Şekil 6.5 AİB ve GİB tabanlı TYPÇY metodlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



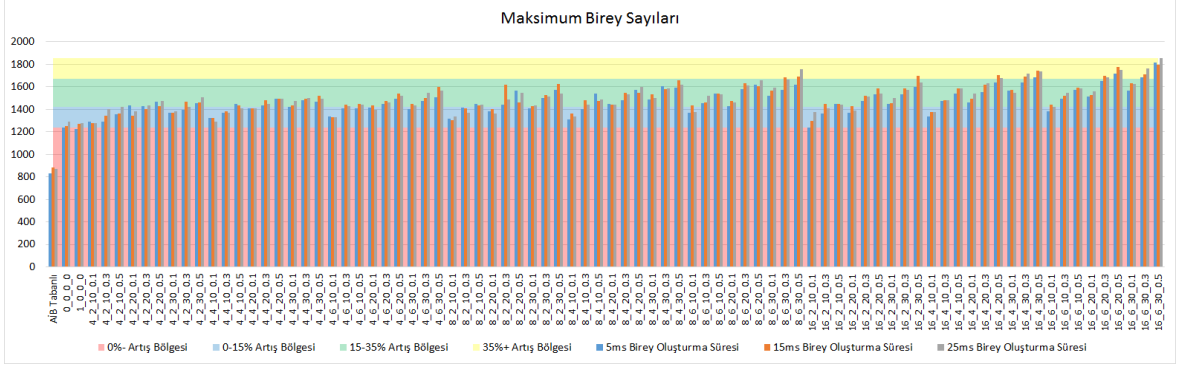
Şekil 6.6 AİB ve GİB tabanlı TYPÇY metodlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



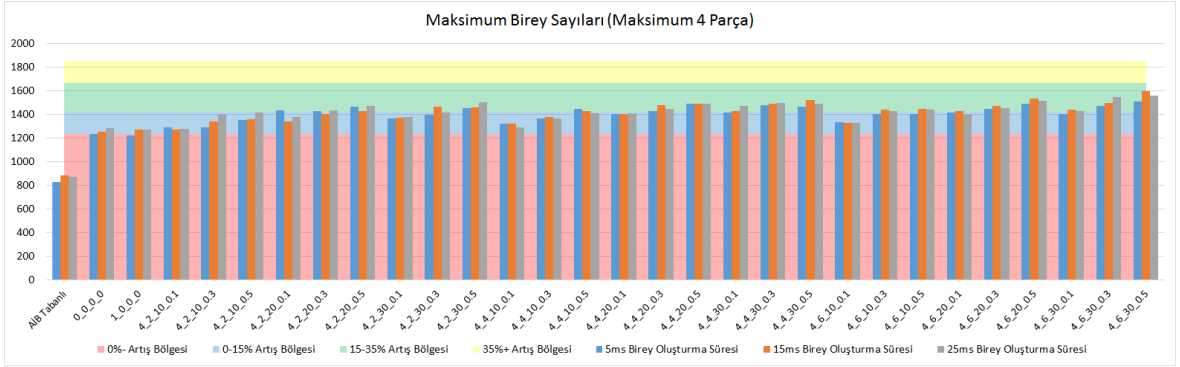
Şekil 6.7 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



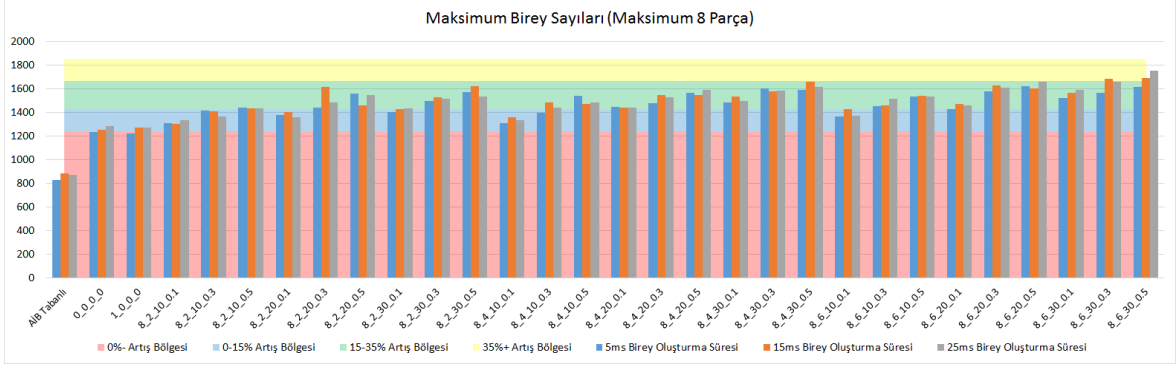
Şekil 6.8 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



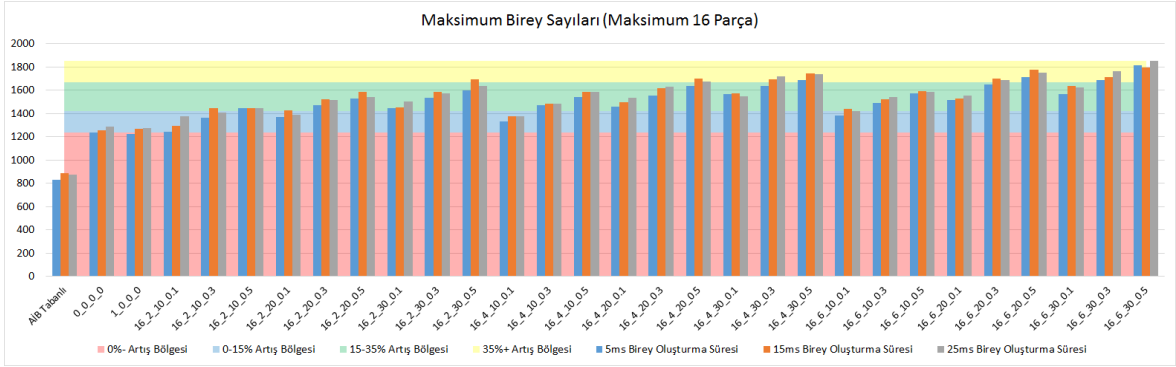
Şekil 6.9 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.10 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.11 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.12 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)

Yine sonuçlardan açıkça görülebildiği üzere ÇYPÇY yöntemi, ÇYPÇY'ler kullanan hemen hemen tüm parametre kombinasyonlarında, GİB tabanlı TYPÇY yönteminden daha yüksek ortalama (%3-53) ve maksimum (%0-47) birey sayılarına ulaşabilmektedir. Özet olarak, her bir benzetim zaman adımında yeni bireyler oluşturmak üzere eşit miktarda süre verilmiş olmasına rağmen, ÇYPÇY yöntemi kullanılan parametre değerlerine bağlı olarak artan birey doygunluğu üst limitleri sayesinde Şekil 6.13'te de görüldüğü gibi daha yoğun kalabalıklar oluşturmayı başarabilmektedir. Bu durumun temel nedeni, ÇYPÇY yönteminde birden fazla yörünge parçası kullanılmasının yanı sıra yörünge parçaları arasındaki geçişlerde yürüme doğrultusu ve hızında değişiklik yapılmasına olanak sağlanarak olası çarpışmasız yol çözüm

uzayının genişletilmesi ve böylelikle yeni çarpışmasız yolların bulunması ihtimalinin artırılmasıdır.



Şekil 6.13 (Sol) GİB tabanlı TYPÇY ve (Sağ) ÇYPCY yöntemleri ile gerçekleştirilen benzetimlerin örnek benzetim alanı görüntüleri

Tüm bunlara ek olarak, kullanılan parametrelerin etkilerinin daha iyi anlaşılmasını sağlamak adına, ÇYPCY yöntemiyle ÇYPCY'ler kullanacak biçimde gerçekleştirilen benzetimlerden elde edilen ortalama ve maksimum birey sayısı verileri istatistiksel olarak analiz edilmiştir. Bu amaçla ortalama ve maksimum birey sayılarının bağımlı değişkenler olarak kullanıldığı ve *max_num_of_segments*, *margin_coef*, *delta_angle*, *delta_speed* ile birey oluşturma için ayrılan süre parametrelerinin bağımsız değişkenler olarak ele alındığı beş yönlü bir MANOVA gerçekleştirilmiştir. Çoklu karşılaştırma analizleri Tukey HSD testi kullanılarak yapılmış ve istatistiksel güven seviyesi %95 olarak belirlenmiştir. *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametrelerinin bağımlı değişkenlerin her ikisi üzerinde de ana etki sahibi olduğu görülmüştür (Çizelge 6.4). Yapılan çoklu karşılaştırma testleri, daha önce kurulan hipotezi doğrular nitelikte, bu parametrelerin herhangi birinde meydana gelen artışın, hem ortalama birey sayısı hem de maksimum birey sayısında artışa neden olduğunu göstermiştir (tüm durumlar için $p \approx 0$). Ayrıca birey oluşturmak için ayrılan süre parametresinin, her iki bağımlı değişken üzerinde de ana etki sahibi olduğu belirlenmiştir (Çizelge 6.4). Çoklu karşılaştırma testlerine göre, birey oluşturmak için her bir zaman adımında 5 ms yerine 15 veya 25 ms ayırmak, ortalama ve maksimum birey sayılarında artışa neden olmuştur (tüm durumlar için $p \approx 0$). Ancak birey oluşturma süresi olarak 15 ms yerine 25 ms kullanmanın, ne ortalama ne de maksimum birey sayısı üzerinde istatistiksel olarak anlamlı bir etkisi görülmüştür (ortalama birey sayısı için $p < 0.965$ ve maksimum birey sayısı için $p < 0.973$). Bağımsız değişkenlerin bağımlı değişkenler üzerindeki etki büyüklükleri (η^2) de Çizelge 6.4'te verilmiştir.

Çizelge 6.4 ÇYPCY yönteminin birey oluşturma sürecinde kullanılan parametrelerin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ort. birey sayısı	delta_angle	2	373.175	0.000	0.282
	delta_speed	2	333.266	0.000	0.252
	max_num_of_segments	2	286.014	0.000	0.216
	margin_coef	2	201.067	0.000	0.152
	Birey oluşturma için ayrılan süre	2	14.316	0.000	0.011
	Hata	232			
Maks. birey sayısı	delta_angle	2	294.573	0.000	0.272
	delta_speed	2	273.481	0.000	0.253
	max_num_of_segments	2	238.324	0.000	0.220
	margin_coef	2	144.572	0.000	0.134
	Birey oluşturma için ayrılan süre	2	14.114	0.000	0.013
	Hata	232			

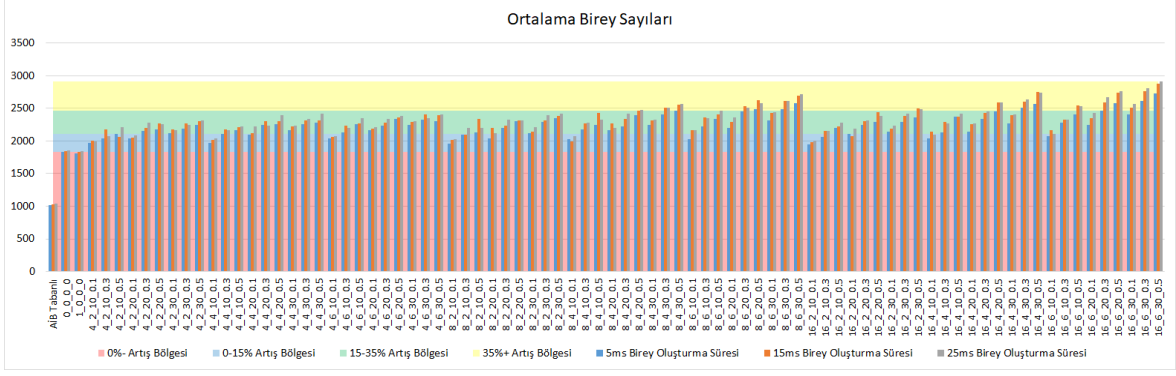
6.2.1.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar

Gerçekleştirilen tüm benzetimlerin ortalama ve maksimum birey sayılarını gösteren grafikler, sırasıyla Şekil 6.14 ve Şekil 6.18’de verilmiştir. Yine sonuçların yorumlanmasını kolaylaştırmak üzere maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, ortalama birey sayısı için sırasıyla Şekil 6.15, Şekil 6.16 ve Şekil 6.17’de, maksimum birey sayısı için sırasıyla Şekil 6.19, Şekil 6.20 ve Şekil 6.21’de verilmiştir. Bu sonuçlara göre, GİB tabanlı TYPÇY yönteminin, AİB tabanlı TYPÇY metoduna kıyasla birey oluşturma için ayrılan süre parametresinin değerine bağlı olarak ortalama birey sayısında %79-80 ve maksimum birey sayısında %95-99 daha yüksek değerlere ulaşabildiği görülmüştür. Aynı şekilde, ÇYPCY yönteminin yalnızca TYPÇY’ler kullanacak şekilde gerçekleştirilen benzetimlerinin, AİB tabanlı TYPÇY metoduna göre, ortalama birey sayısında %78-79 ve maksimum birey sayısında %89-93 daha yüksek değerlere ulaşabildiği belirlenmiştir. GİB tabanlı TYPÇY ve ÇYPCY yöntemleriyle yalnızca TYPÇY’ler kullanacak şekilde gerçekleştirilen benzetimlerde, hem ortalama hem de maksimum birey sayıları arasında GİB tabanlı TYPÇY yöntemi lehine %1-3 gibi küçük farklar bulunmaktadır. GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin yalnızca TYPÇY’ler kullanacak şekilde, birey oluşturma için ayrılan süre parametresinin yukarıda belirtilen farklı değerleri için tekrarlanan benzetimlerinden elde edilen ortalama ve maksimum birey sayıları

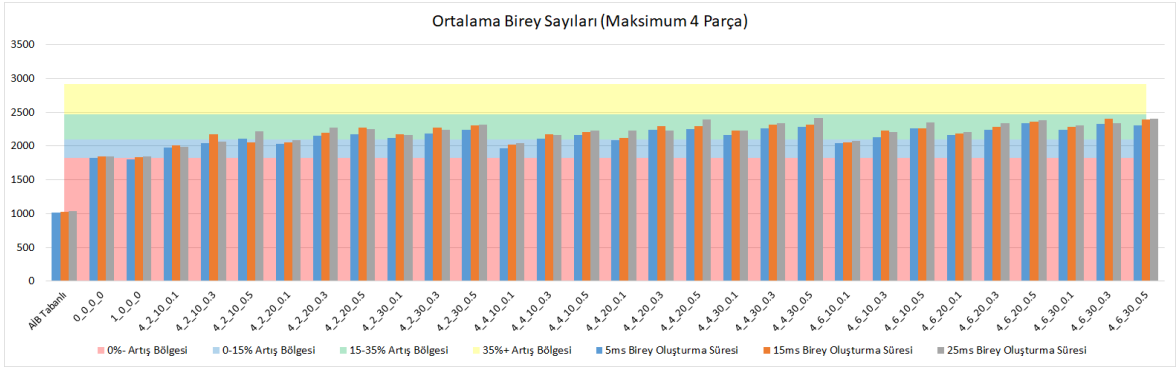
üzerinde %95 güven seviyesinde iki yönlü bir MANOVA gerçekleştirilmiştir. Ortalama ve maksimum birey sayıları bağımlı değişkenler olarak kullanılırken gezinim yöntemi ve birey oluşturma için ayrılan süre parametreleri bağımsız değişkenler olarak ele alınmıştır. Çoklu karşılaştırma analizleri ise Tukey HSD testi kullanılarak gerçekleştirilmiştir. Birey sayıları üzerinde gerçekleştirilen bu istatistiksel analizler sonucunda, hem ortalama ($F(1, 26) = 14.576, p < 0.001$) hem de maksimum birey sayıları ($F(1, 26) = 38.393, p \approx 0$) arasındaki bu farkların istatistiksel açıdan anlamlı olduğu tespit edilmiştir (Çizelge 6.5). Bu küçük farklılıkların, GİB tabanlı TYPÇY yönteminin, yalnızca TYPÇY'ler kullanan ÇYPCY yöntemine göre nispeten daha düşük birey oluşturma maliyetine sahip olmasından dolayı (6.3.1 Birey Oluşturma Maliyetlerinin Karşılaştırılması) her bir zaman adımında birey oluşturma için ayrılan süre içerisinde, daha fazla birey oluşturma denemesi yapabilmekten kaynaklandığı düşünülmektedir. Diğer taraftan, sadece TYPÇY'ler kullanan bu gezinim yöntemleriyle gerçekleştirilen benzetimlerde, her bir zaman adımı içerisinde birey oluşturma için ayrılan sürenin artırılmasının, ortalama birey sayısı ($F(2, 26) = 6.468, p < 0.005$) ve maksimum birey sayısı ($F(2, 26) = 6.478, p < 0.005$) üzerinde istatistiksel açıdan anlamlı etkisi olduğunu söylemek mümkündür (Çizelge 6.5). Çoklu karşılaştırma testlerine göre, birey oluşturmak için her bir zaman adımında 5 ms yerine 25 ms ayırmak ortalama ve maksimum birey sayılarının her ikisinde de artışa neden olurken (ortalama birey sayısı için $p < 0.004$ ve maksimum birey sayısı için $p < 0.006$) 5 ms yerine 15 ms ayırmak yalnızca maksimum birey sayısında artışa neden olmuştur ($p < 0.029$).

Çizelge 6.5 GİB tabanlı TYPÇY ve yalnızca TYPÇY'ler kullanan ÇYPCY yöntemleri ile birey oluşturma için ayrılan süre parametresinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)

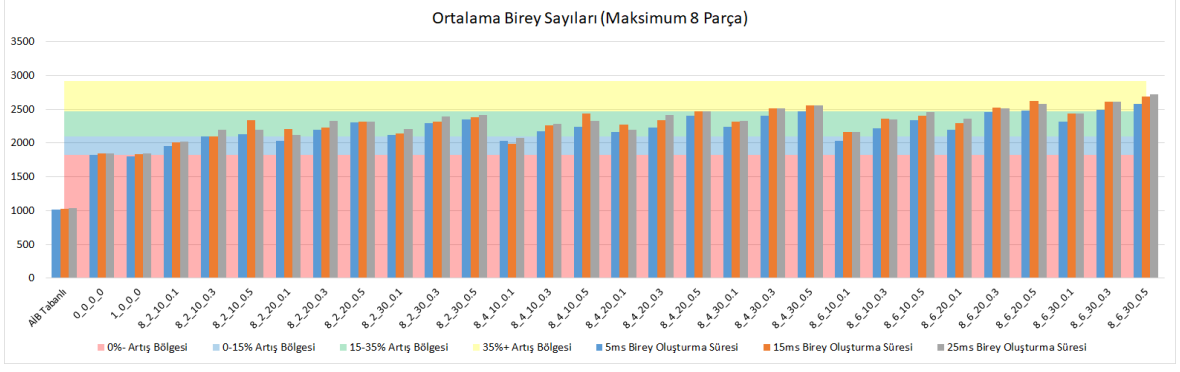
Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Gezinim yöntemi	1	14.576	0.001	0.272
	Birey oluşturma için ayrılan süre	2	6.468	0.005	0.242
	Hata	26			
Maksimum birey sayısı	Gezinim yöntemi	1	38.393	0.000	0.496
	Birey oluşturma için ayrılan süre	2	6.478	0.005	0.168
	Hata	26			



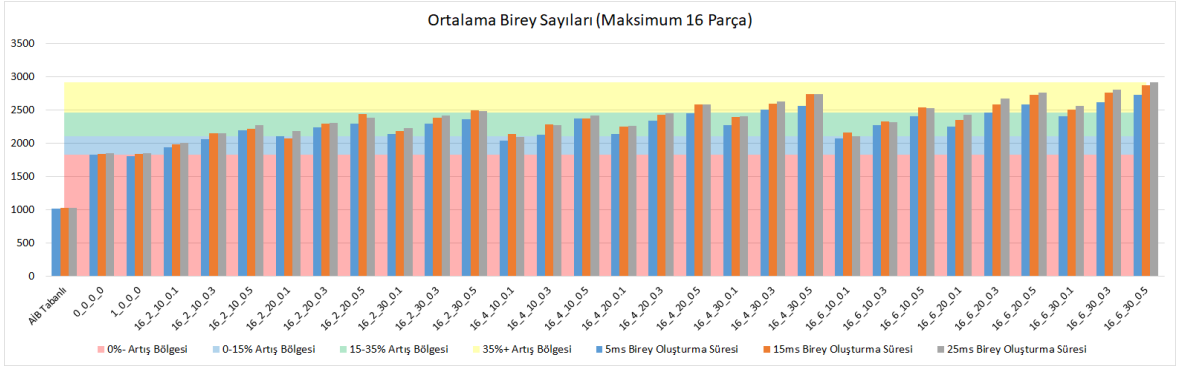
Şekil 6.14 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



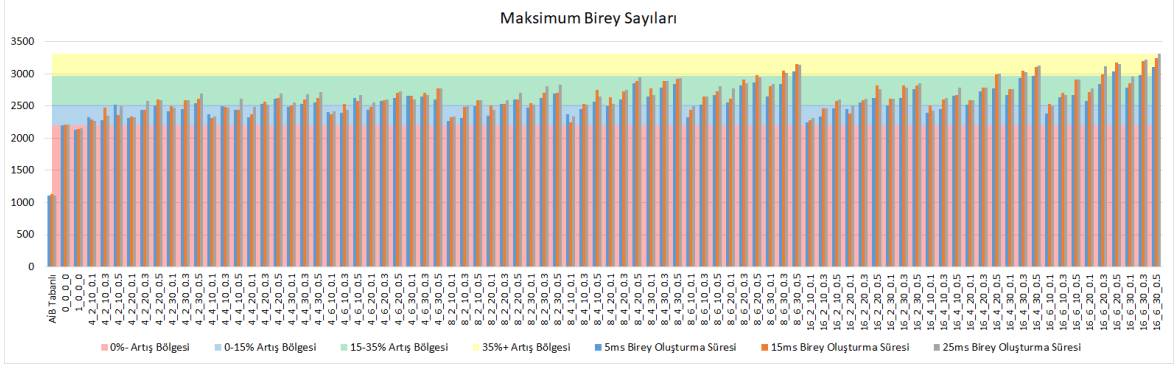
Şekil 6.15 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



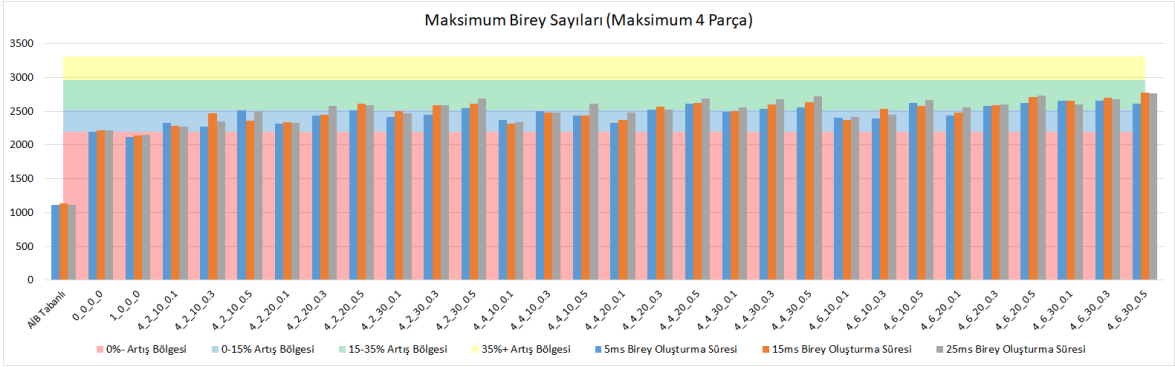
Şekil 6.16 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



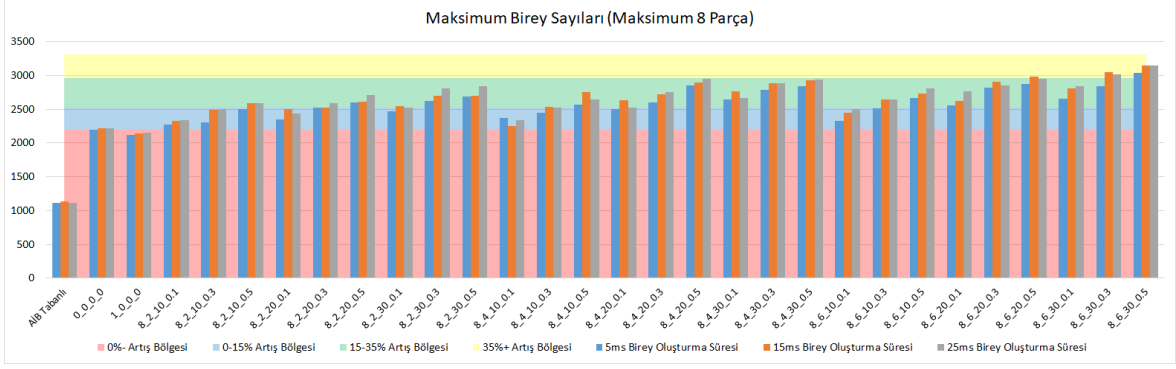
Şekil 6.17 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



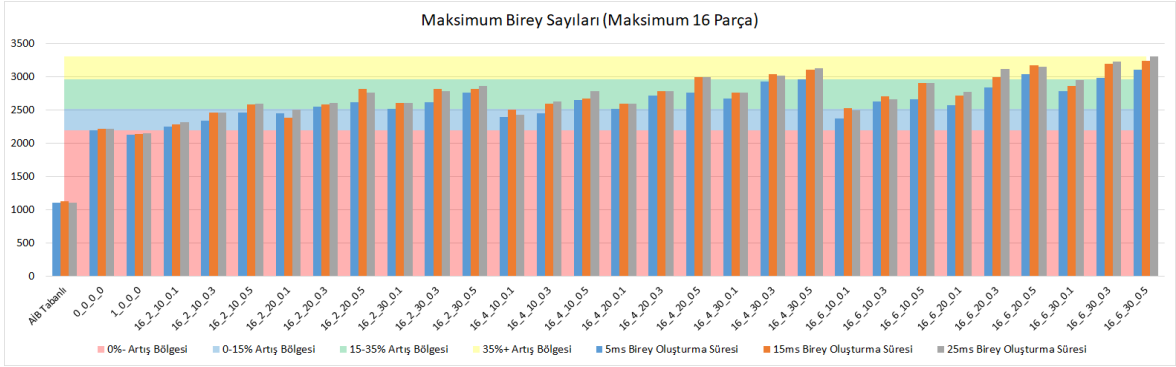
Şekil 6.18 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.19 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.20 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.21 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)

Yine sonuçlardan açıkça görülebildiği üzere ÇYPCY yöntemi, ÇYPCY'ler kullanan tüm parametre kombinasyonlarında, GİB tabanlı TYPÇY yönteminden daha yüksek ortalama (%6-58) ve maksimum (%1-50) birey sayılarına ulaşabilmektedir. Özetle, her bir benzetim zaman adımında eşit birey oluşturma süreleri ayrılmış olmasına rağmen, birden çok yörünge parçası kullanmakla kalmayıp bu yörünge parçaları arasındaki geçişlerde yürüme doğrultusu ve hızında değişiklik yapılmasına olanak sağlayarak olası çözüm uzayını genişleten ve böylelikle yeni çarpışmasız yolların bulunması ihtimalinin artıran ÇYPCY yöntemi, birey doygunluğu üst limitlerindeki artışa bağlı olarak daha yoğun kalabalıklar oluşturmayı başarabilmektedir.

Bunların yanı sıra, kullanılan parametrelerin etkilerinin daha iyi anlaşılmasını sağlamak adına, ÇYPCY yöntemiyle ÇYPCY'ler kullanacak biçimde gerçekleştirilen benzetimlerden elde edilen ortalama ve maksimum birey sayısı verileri istatistiksel olarak analiz edilmiştir. Bu amaçla ortalama ve maksimum birey sayılarının bağımlı değişkenler olarak kullanıldığı ve *max_num_of_segments*, *margin_coef*, *delta_angle*, *delta_speed* ile birey oluşturma için ayrılan süre parametrelerinin bağımsız değişkenler olarak ele alındığı beş yönlü bir MANOVA gerçekleştirilmiştir. Çoklu karşılaştırma analizleri Tukey HSD testi kullanılarak gerçekleştirilmiş ve istatistiksel güven seviyesi %95 olarak belirlenmiştir. *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametrelerinin bağımlı değişkenlerin her ikisi üzerinde de ana etki sahibi olduğu görülmüştür (Çizelge 6.6). Yapılan çoklu karşılaştırma testlerine göre, daha önce kurulan hipoteze ve birey yarıçapının 0.5 m alındığı deneylerin sonuçlarına paralel olarak, bu parametrelerin herhangi birinde meydana gelen artışın, hem ortalama birey sayısı hem de maksimum birey sayısında artışa neden olduğunu söylemek mümkündür (tüm durumlar için $p \approx 0$). Ek olarak, birey oluşturmak için ayrılan süre parametresinin, her iki bağımlı değişken üzerinde de ana etki sahibi olduğu tespit edilmiştir (Çizelge 6.6). Çoklu karşılaştırma testlerine göre, birey oluşturmak için her bir zaman adımında 5 ms yerine 15 veya 25 ms ayırmak, ortalama ve maksimum birey sayılarında artışa neden olmuştur (tüm durumlar için $p \approx 0$). Diğer taraftan, birey oluşturma süresi olarak 15 ms yerine 25 ms kullanmanın, ortalama veya maksimum birey sayısı üzerinde istatistiksel olarak anlamlı bir etkisi belirlenememiştir (ortalama birey sayısı için $p < 0.195$ ve maksimum birey sayısı için $p < 0.207$). Bağımsız değişkenlerin bağımlı değişkenler üzerindeki etki büyüklükleri (η^2) de Çizelge 6.6'da verilmiştir.

Çizelge 6.6 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ort. birey sayısı	delta_speed	2	317.619	0.000	0.278
	delta_angle	2	287.474	0.000	0.251
	margin_coef	2	212.264	0.000	0.186
	max_num_of_segments	2	155.888	0.000	0.136
	Birey oluşturma için ayrılan süre	2	54.669	0.000	0.048
	Hata	232			
Maks. birey sayısı	delta_angle	2	258.051	0.000	0.263
	delta_speed	2	227.964	0.000	0.232
	margin_coef	2	176.804	0.000	0.180
	max_num_of_segments	2	164.754	0.000	0.168
	Birey oluşturma için ayrılan süre	2	38.902	0.000	0.040
	Hata	232			

Öngörülebileceği üzere bireylerin yarıçapındaki azalmaya bağlı olarak benzetim alanı üzerinde kapladıkları sınırlayıcı alan da azaldığından, aynı benzetim alanı üzerinde çok daha fazla birey oluşturmak mümkün hale gelmektedir. Sonuçlar incelendiğinde genel olarak yarıçapı 0.3 m olan bireyler ile gerçekleştirilen benzetimlerin, yarıçapı 0.5 m olan bireyler ile gerçekleştirilen benzetimlerden daha yüksek birey sayılarına ulaşabildiği göze çarpmaktadır. AİB tabanlı TYPÇY yönteminde bu artış, birey oluşturma için ayrılan sürenin farklı değerlerine bağlı olarak ortalama birey sayısı için %29-30 iken maksimum birey sayısı için %28-34 arasındadır. Diğer taraftan, çok daha belirleyici bir birey oluşturma yaklaşımı kullanan GİB tabanlı TYPÇY yönteminde birey sayılarındaki bu artış, ortalama birey sayısı için %73-77 ve maksimum birey sayısı için %72-78 olmakla birlikte ÇYPÇY yönteminin yalnızca TYPÇY'ler kullanan benzetimlerinde, ortalama birey sayısı için %77-78 ve maksimum birey sayısı için %69-74 civarındadır. Benzer biçimde, ÇYPÇY yönteminin ÇYPÇY'ler kullanan benzetimlerinde ise parametrelerin farklı kombinasyonlarına bağlı olarak söz konusu bu artışın miktarı, ortalama birey sayısı için %66-88 ve maksimum birey sayısı için %56-90 aralıklarında değişmektedir.

6.2.2. Çözüm Seçim Yöntemlerinin İrdelenmesi

Daha önce belirtildiği üzere ÇYPÇY yöntemi, her bir yörünge parçası arama süreci boyunca olası çözümler arasından en yüksek yürüme hızına sahip olan çözümlerden birini seçerek yörünge parçası oluşturmak için kullanılmaktadır. Bu karakteristiğinin arkasında yatan mantık

şudur: Her bir yörünge parçasına olası çözümler arasında mümkün olan en yüksek hızın atanması sayesinde bireyler benzetim alanı üzerinde daha kısa zaman aralıkları boyunca kişisel alan işgal edecektir. Böylelikle benzetimde var olan bireyler daha hızlı hareket ederek benzetim alanı üzerinde yeni bireyler için daha hızlı yer açacağından birey oluşturma denemeleri sonucunda yeni bireylerin oluşturulması daha muhtemel hale gelecektir. Tam tersine Fruin [139] tarafından belirtildiği gibi daha yüksek hızlarla yürüyen bireyler benzetim alanı üzerinde daha geniş kişisel alanlar gerektirerek birey yoğunluğu doygunluk noktasının azalmasına ve bunun sonucu olarak benzetimi yapılabilen birey sayısında düşüşe de neden olabilir.

Olası çözümler arasından maksimum hızlı olanın seçilmesi kararının geçerliliğini araştırmak üzere bir takım yeni deneyler gerçekleştirilmiştir. Bu amaçla yörünge parçası oluşturma sürecinde üç farklı çözüm seçim yöntemi ele alınmıştır. Bunlardan ilki yukarıda bahsedildiği gibi olası çözümler arasından en yüksek yürüme hızına sahip birini seçmektedir. İkincisi ilk yöntemin tam aksine olası çözümler arasından en düşük yürüme hızına sahip birini seçmektedir. Sonuncusu ise ilk iki yöntemin sonuçlarının kıyaslanacağı bir taban oluşturmak adına olası çözümler arasından birini tamamen rastgele seçmektedir. Önceki deneylerde olduğu gibi *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametrelerinin aynı değerleri kullanılırken birey oluşturma için harcanan süre olarak sadece 25 ms değeri kullanılmıştır. Bu kararın verilmesinin nedeni, en fazla sayıda birey oluşturma denemesinin gerçekleştirilmesine olanak tanımanın kullanılan çözüm seçim yöntemlerinin maksimum potansiyellerini açığa çıkararak kendileri için mümkün olan en yüksek ortalama ve maksimum birey sayılarına ulaşmalarını sağlayacağını düşünülmesidir. Bir kenarının uzunluğu 128 m olan kare biçiminde bir benzetim alanı üzerinde, bu parametrelerin her bir kombinasyonu için, her bir çözüm seçim yöntemi ile ayrı ayrı 5 dakikalık gerçek zamanlı benzetimler çalıştırılmıştır. Başlangıçta boş olan benzetim alanını olabildiğince fazla birey oluşturarak doldurmaya çalışan bu benzetimlerde, her saniyede benzetimde yer alan birey sayıları ölçülerek benzetimin sonunda ortalama ve maksimum birey sayıları hesaplanmıştır. Deneyler, birey yarıçapının 0.5 m ve 0.3 m olmak üzere iki farklı değeri için iki farklı grup olarak tekrarlanmıştır.

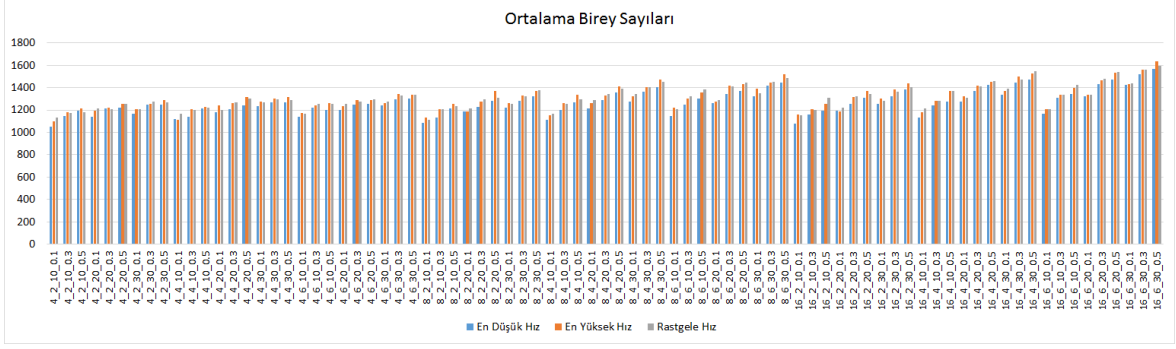
6.2.2.1. Birey Yarıçapının 0.5 m Değeri için Sonuçlar

Kullanılan üç farklı çözüm seçim yöntemi ile elde edilen ortalama ve maksimum birey sayıları, sırasıyla Şekil 6.22 ve Şekil 6.26'da gösterilmiştir. Bunun yanı sıra, sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim

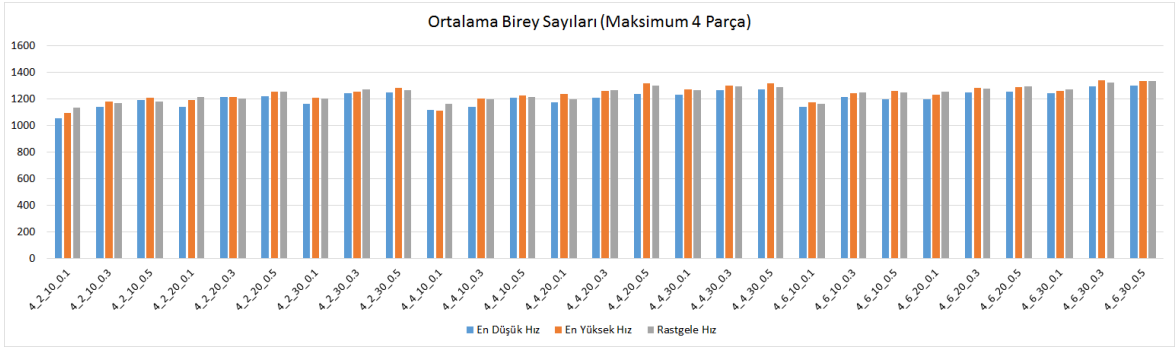
gruplarına göre bölümlenmiş grafikler, ortalama birey sayısı için sırasıyla Şekil 6.23, Şekil 6.24 ve Şekil 6.25'te, maksimum birey sayısı için sırasıyla Şekil 6.27, Şekil 6.28 ve Şekil 6.29'da verilmiştir. Bu sonuçlardan da görülebildiği üzere rastgele veya en yüksek yürüme hızına göre bir çözüm seçmek, en düşük yürüme hızına sahip çözümlerden birini seçmeye göre hem ortalama birey sayıları (ortalama olarak rastgele için %3.52 ve en yüksek hız için %3.57) hem de maksimum birey sayılarında (ortalama olarak rastgele için %4.88 ve en yüksek hız için %4.57) daha yüksek sonuçlar üretmektedir. Ancak en yüksek hız ve rastgele çözüm seçim yöntemleri arasındaki ortalama fark, ortalama birey sayısı için %1.32'den, maksimum birey sayısı için ise %1.90'dan daha düşüktür. Yine de daha derin bilgi sağlaması adına elde edilen bu benzetim verileri üzerinde %95 güven seviyesinde tek yönlü bir MANOVA gerçekleştirilmiştir. Ortalama ve maksimum birey sayılarının bağımlı değişken olarak kullanıldığı bu analizlerde çözüm seçim yöntemi bağımsız değişken olarak ele alınmış ve çoklu karşılaştırma analizleri Tukey HSD testi kullanılarak yapılmıştır. Yapılan bu analizler sonucunda, kullanılan çözüm seçim yönteminin, ortalama birey sayıları ($F(2, 240) = 4.728, p < 0.010$) ve maksimum birey sayıları ($F(2, 240) = 9.244, p \approx 0$) üzerinde ana etkiye sahip olduğu tespit edilmiştir (Çizelge 6.7). Çoklu karşılaştırma testleri, rastgele ve en yüksek hız yöntemlerinin, hem ortalama birey sayılarında (rastgele için $p < 0.021$ ve en yüksek hız için $p < 0.024$) hem de maksimum birey sayılarında (rastgele için $p < 0.001$ ve en yüksek hız için $p \approx 0$) en düşük hız yönteminden daha fazla birey oluşturabildiğini göstermiştir. Buna karşın, rastgele bir olası çözüm seçmek yerine en yüksek hıza sahip olası çözümlerden birini seçmenin, ne ortalama birey sayılarında ($p < 0.998$) ne de maksimum birey sayılarında ($p < 0.997$) istatistiksel açıdan anlamlı bir fark oluşturmadığı görülmüştür.

Çizelge 6.7 ÇYPÇY yönteminde kullanılan çözüm seçim yöntemlerinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)

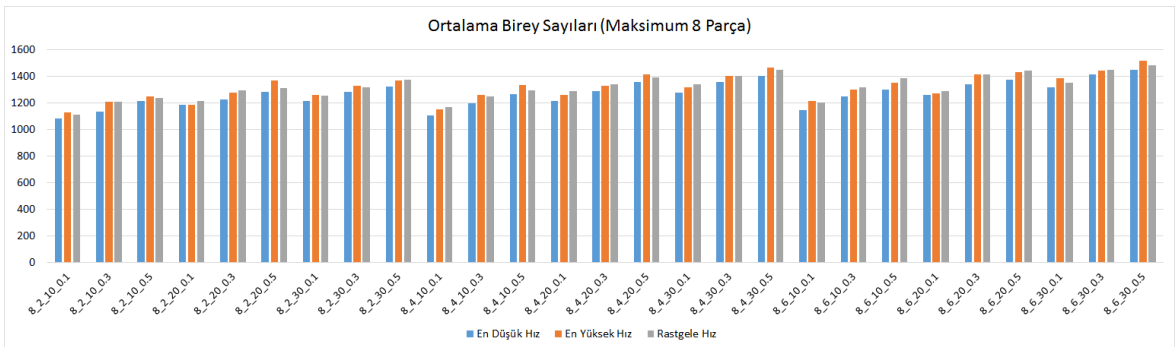
Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Çözüm seçim yöntemi	2	4.728	0.010	0.038
	Hata	240			
Maksimum birey sayısı	Çözüm seçim yöntemi	2	9.244	0.000	0.072
	Hata	240			



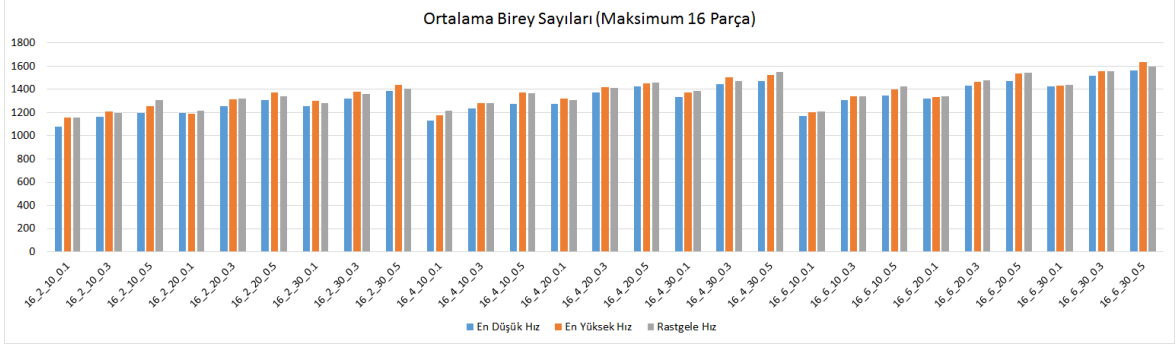
Şekil 6.22 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



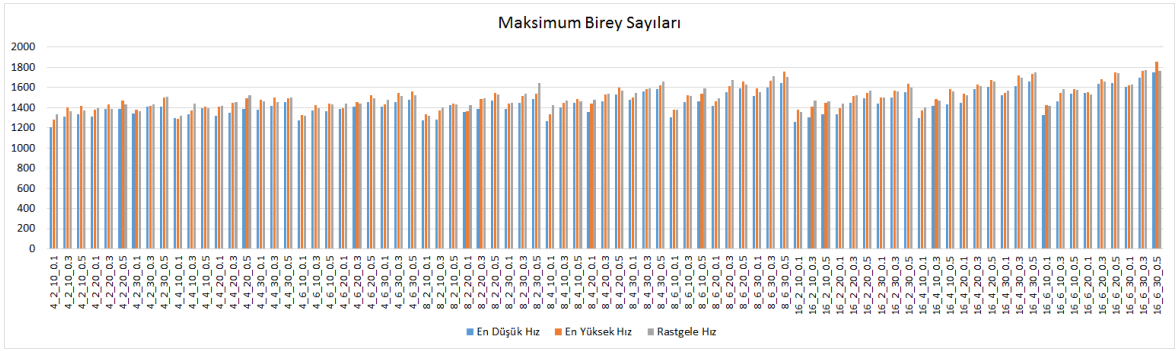
Şekil 6.23 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



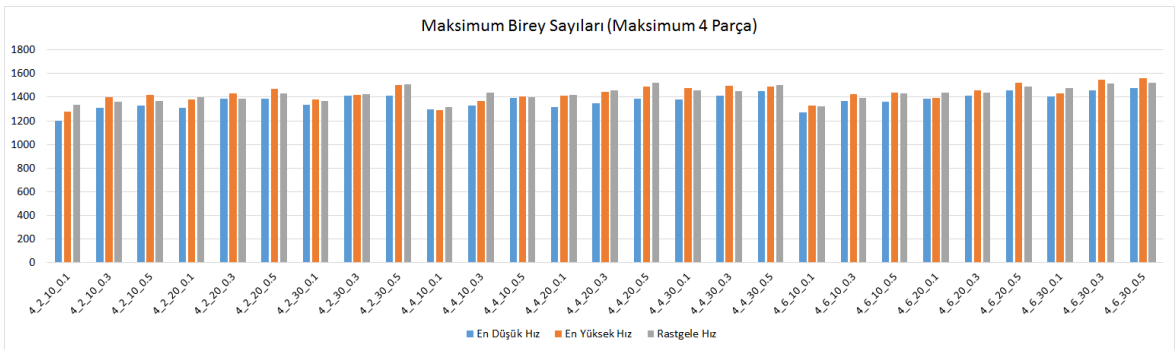
Şekil 6.24 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



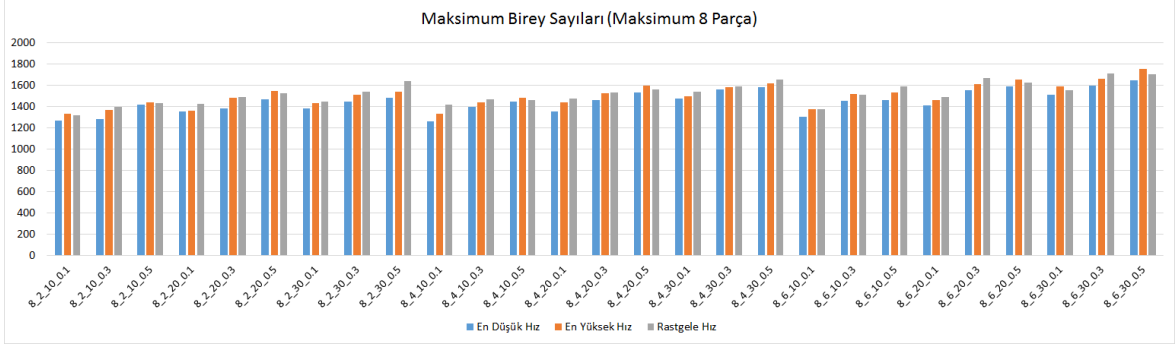
Şekil 6.25 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.5 m$)



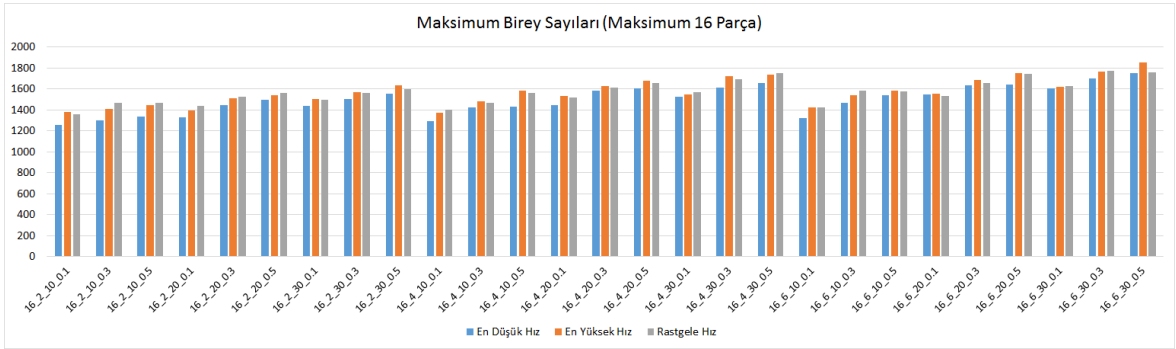
Şekil 6.26 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.27 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.28 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.29 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.5 m$)

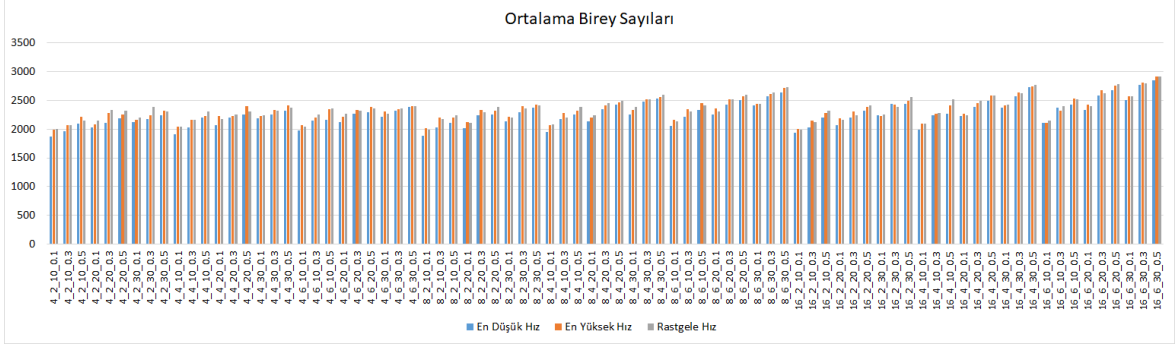
6.2.2.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar

Çözüm seçim yönteminin ortalama ve maksimum birey sayıları, sırasıyla Şekil 6.30 ve Şekil 6.34'te gösterilmiştir. Ayrıca sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, ortalama birey sayısı için sırasıyla Şekil 6.31, Şekil 6.32 ve Şekil 6.33'te, maksimum birey sayısı için sırasıyla Şekil 6.35, Şekil 6.36 ve Şekil 6.37'de verilmiştir. Elde edilen bu sonuçlara göre, en yüksek yürüme hızına göre bir çözüm seçmek, en düşük yürüme hızına sahip çözümlerden birini seçmeye göre hem ortalama birey sayıları (ortalama olarak %3.56) hem de maksimum birey sayılarında (ortalama olarak %4.68) daha yüksek sonuçlar üretmektedir. Aynı şekilde, rastgele bir çözüm seçmek, en düşük yürüme hızına sahip çözümlerden birini seçmeye göre hem ortalama birey sayıları (ortalama olarak %3.82) hem de maksimum birey

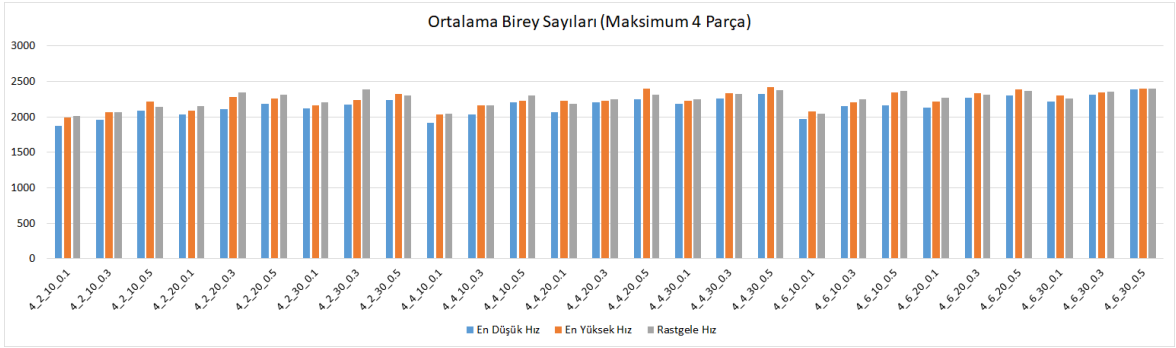
sayılarında (ortalama olarak %4.97) daha yüksek sonuçlar üretmektedir. Diğer taraftan, en yüksek hız ve rastgele çözüm seçim yöntemleri arasındaki ortalama fark, ortalama birey sayısı için %1.46'dan, maksimum birey sayısı için ise %2.04'ten daha düşüktür. Daha derin bilgi sahibi olmak adına elde edilen bu benzetim verileri üzerinde %95 güven seviyesinde tek yönlü bir MANOVA gerçekleştirilmiştir. Ortalama ve maksimum birey sayılarının bağımlı değişken olarak kullanıldığı bu analizlerde çözüm seçim yöntemi bağımsız değişken olarak ele alınmış ve çoklu karşılaştırma analizleri Tukey HSD testi kullanılarak yapılmıştır. Yapılan bu analizler sonucunda, kullanılan çözüm seçim yönteminin, ortalama birey sayıları ($F(2, 240) = 4.450, p < 0.013$) ve maksimum birey sayıları ($F(2, 240) = 7.411, p < 0.001$) üzerinde ana etkiye sahip olduğu görülmüştür (Çizelge 6.8). Çoklu karşılaştırma testleri, en yüksek hız yönteminin, hem ortalama birey sayılarında ($p < 0.036$) hem de maksimum birey sayılarında ($p < 0.004$) en düşük hız yönteminden daha fazla birey oluşturabildiğini göstermiştir. Aynı şekilde, rastgele yönteminin hem ortalama birey sayılarında ($p < 0.022$) hem de maksimum birey sayılarında ($p < 0.002$) en düşük hız yönteminden daha fazla birey oluşturabildiği tespit edilmiştir. Buna rağmen, rastgele bir olası çözüm seçmek yerine en yüksek hıza sahip olası çözümlerden birini seçmenin, ne ortalama birey sayılarında ($p < 0.980$) ne de maksimum birey sayılarında ($p < 0.976$) istatistiksel açıdan anlamlı bir fark oluşturmadığı belirlenmiştir.

Çizelge 6.8 ÇYPCY yönteminde kullanılan çözüm seçim yöntemlerinin ortalama ve maksimum birey sayıları üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)

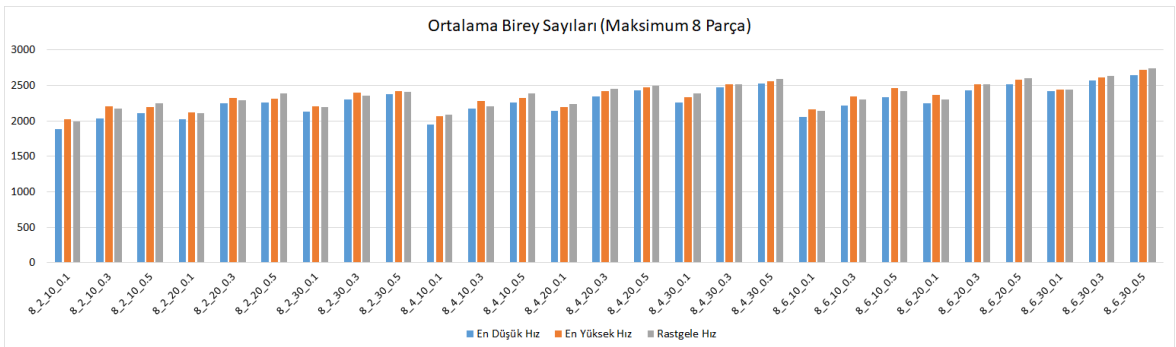
Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Ortalama birey sayısı	Çözüm seçim yöntemi	2	4.450	0.013	0.036
	Hata	240			
Maksimum birey sayısı	Çözüm seçim yöntemi	2	7.411	0.001	0.058
	Hata	240			



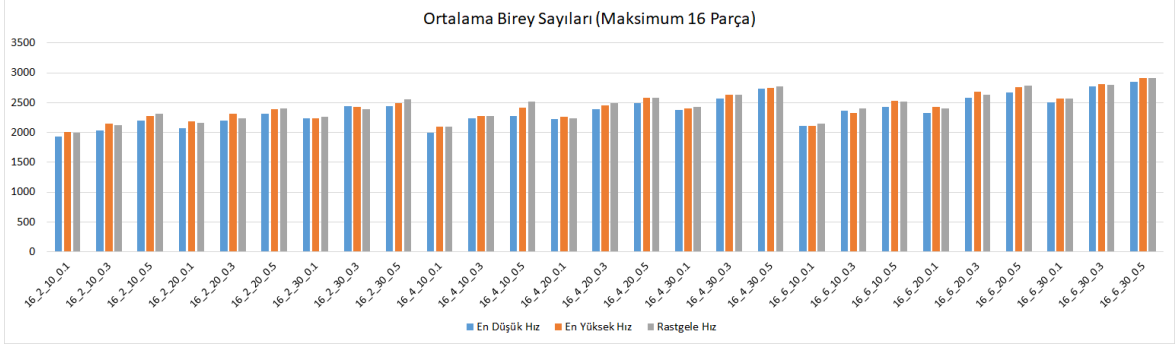
Şekil 6.30 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



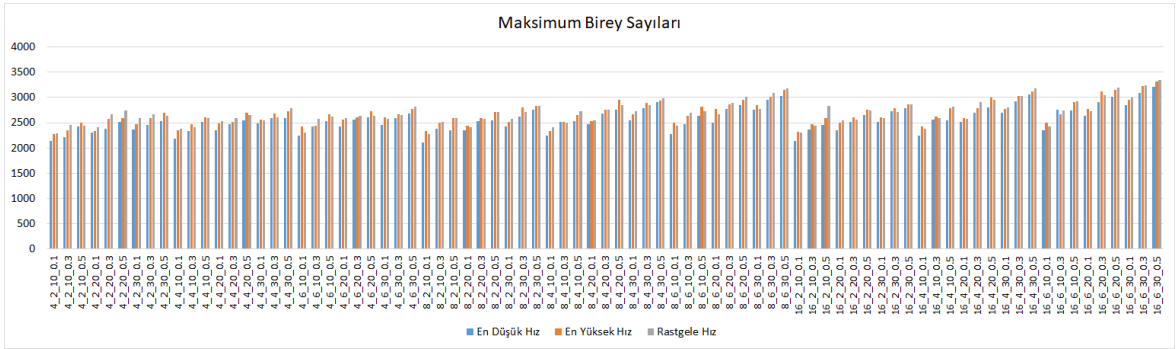
Şekil 6.31 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



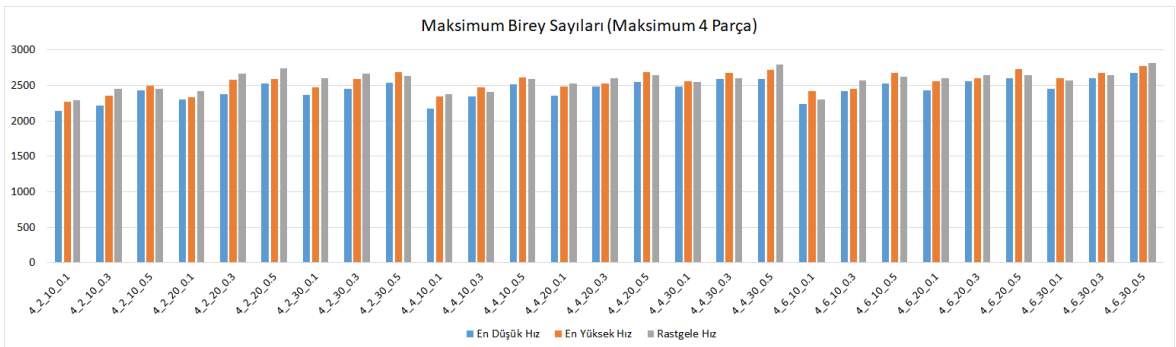
Şekil 6.32 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



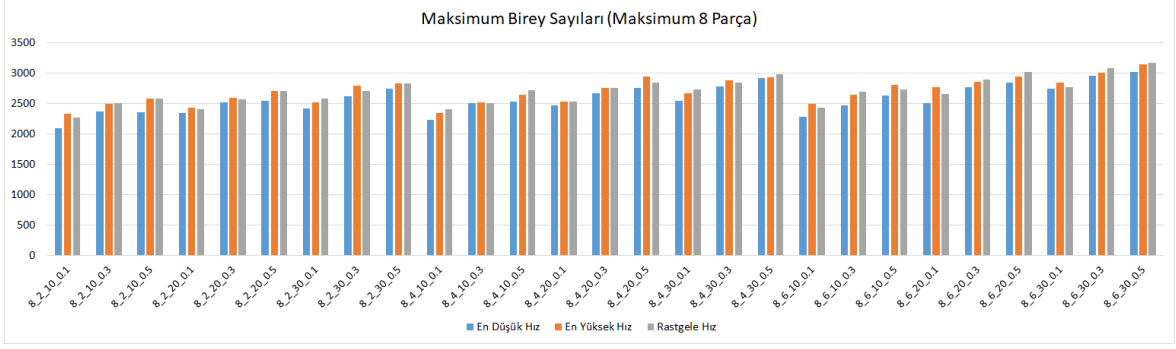
Şekil 6.33 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey sayıları ($r = 0.3 m$)



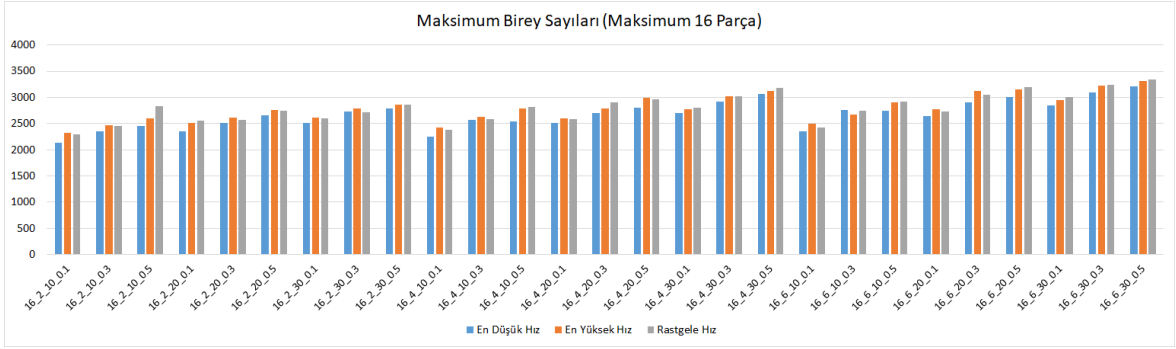
Şekil 6.34 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.35 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.36 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.37 Üç farklı çözüm seçme yöntemini kullanarak gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için maksimum birey sayıları ($r = 0.3 m$)

Daha önce de belirtildiği üzere yarıçapı $0.3 m$ olan bireyler ile gerçekleştirilen benzetimler aynı benzetim alanı üzerinde yarıçapı $0.5 m$ olan bireyler ile gerçekleştirilen benzetimlere kıyasla çok daha yüksek birey sayılarına ve dolayısıyla birey yoğunluklarına ulaşabilmektedir. Buna göre, yarıçapı $0.5 m$ olan bireyler kullanmak yerine yarıçapı $0.3 m$ olan bireyler kullanmak görevlendirilen farklı çözüm seçim yöntemlerine ve birey oluşturma sürecindeki parametrelerin farklı kombinasyonlarına bağlı olarak ortalama birey sayısını %69-94 ve maksimum birey sayısını %66-94 arasında yükseltmektedir.

6.3. Geliştirilen Gezinim Yöntemlerinin Maliyet Analizi

Geliştirilen gezinim yöntemlerinde gerçekleştirilen hesaplamalar iki gruba ayrılabilir. Bunlardan ilki kalabalığı meydana getiren bireylerin oluşturulması için, ikincisi ise bu bireylerin benzetim içerisinde geziniminin sağlanması için gerçekleştirilmektedir. Bu

doğrultuda tez kapsamında geliştirilen üç gezinim yönteminin her iki gruptaki hesaplamalar için harcadıkları ortalama süreleri ölçen testler gerçekleştirilmiştir. Aynı testler, alanda mevcut en güncel ve en yaygın kullanıma sahip birey tabanlı gezinim yöntemlerinden biri olan Reciprocal Velocity Obstacles (RVO) [64] için de yapılarak tüm yöntemlerin kapsamlı bir karşılaştırmasının yapılmasına olanak sağlanmıştır. Ayrıca tüm bu yöntemlerin bellek gereksinimleri de incelenerek birbirlerine göre kıyaslaması yapılmıştır.

6.3.1. Birey Oluşturma Maliyetlerinin Karşılaştırılması

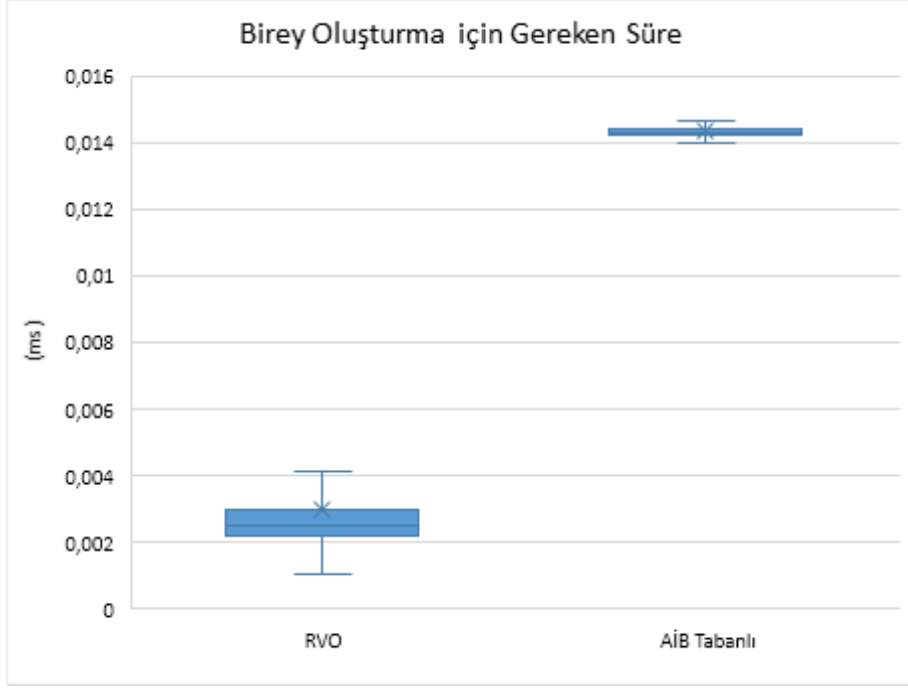
İlk grubu oluşturan hesaplamaların maliyetini ifade etmek üzere birey oluşturma denemeleri için harcanan ortalama sürenin kullanılması oldukça uygundur. Bu amaçla bir kenarının uzunluğu 128 *m* olan kare biçiminde bir benzetim alanı üzerinde tüm yöntemler için ayrı ayrı 5 dakikalık benzetimler çalıştırılmıştır. Başlangıçta boş olan benzetim alanını olabildiğince fazla birey oluşturarak doldurmaya çalışan bu benzetimler çevrimdışı olarak gerçekleştirilmiştir. Zaman adımlarının arasında 1 *ms* süre olacak biçimde ayarlanan bu benzetimlerde her bir zaman adımında birey oluşturma için ayrılan süre sınırı kaldırılmış, ancak her bir zaman adımı içerisinde yalnızca tek bir birey oluşturma denemesi yapılmasına izin verilmiştir. Benzetim boyunca gerçekleştirilecek başarılı ve başarısız tüm birey oluşturma denemelerinin tamamlanma süreleri ölçülerek benzetim sonunda birey oluşturma denemeleri için harcanan ortalama süre hesaplanmıştır. ÇYPCY yönteminde *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametreleri için önceki deneylerle aynı değerler kullanılmış ve benzer biçimde bu parametre değerlerinin farklı her bir kombinasyonu için ayrı bir benzetim çalıştırılmıştır. Birey yarıçapının 0.5 *m* ve 0.3 *m* olmak üzere iki farklı değeri için deneyler iki farklı grup halinde gerçekleştirilmiştir.

Değinilmesi gereken bir diğer önemli nokta ise RVO yönteminde maksimum birey sayısı doygunluğunun çok daha yüksek olmasıdır. Yapılan testlerde RVO yöntemi, birey oluşturmak üzere kare şeklindeki benzetim alanının rastgele kenarlarından birisi üzerinde yer alan rastgele bir başlangıç pozisyonu seçerek bu nokta üzerinde oluşturulacak bireyi temsil eden *r* yarıçaplı diskin var olan hiçbir bireyin diski ile kesişmediğinden emin olduktan sonra bireyi benzetime dâhil edecek biçimde ayarlanmıştır. Ancak birey oluşturma sürecinde gerçekleştirilen bu disk kesişim testleri, bireylerin pozisyonları (disklerin merkezleri) arasındaki basit uzaklık hesaplamalarına karşılık gelmekle birlikte birey sayısı üst sınırı için önemli bir engel teşkil etmemektedir. Diğer yandan, RVO yöntemi her benzetim zaman adımında tüm bireyler için yeni birer hız vektörü hesaplamakla kalmayıp, ihtiyaca göre bireylerin hızını sıfırlayarak onları durdurma ve bu şekilde bekletme olanağına da sahip

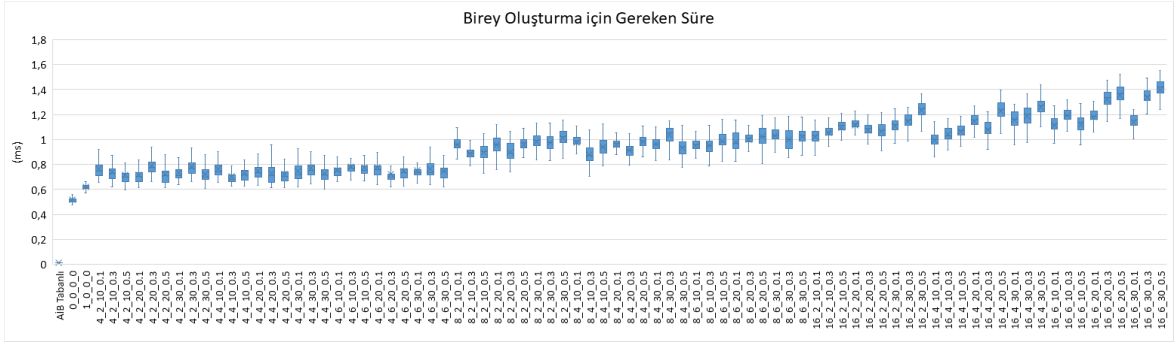
olduğu için çok daha yüksek birey yoğunluklarına ulaşabilmektedir. Her benzetim zaman adımında bir birey oluşturma denemesi yapılması durumunda RVO yöntemi rakiplerine göre daha fazla birey oluşturmak için uğraşırken çok daha fazla hesaplama yapmak suretiyle dezavantajlı duruma düşecektir. Bunun önüne geçmek adına RVO yönteminde birey sayısı üst sınırı, birey yarıçapının $0.5\ m$ olduğu deney grubu için 1600 ve birey yarıçapının $0.3\ m$ olduğu deney grubu için ise 2900 olarak belirlenmiştir. Bu sayede RVO yönteminin, birey sayıları bu iki değeri geçmeyecek şekilde birey oluşturarak sabit birey yoğunluklarına sahip olması sağlanmıştır. Bu değerler, bir önceki bölümde iki farklı yarıçap değeri için elde edilen sonuçlara göre ÇYPCY yönteminin ulaşabildiği maksimum ortalama birey sayısı değerlerini geçmeyecek biçimde seçilmiştir.

6.3.1.1. Birey Yarıçapının $0.5\ m$ Değeri için Sonuçlar

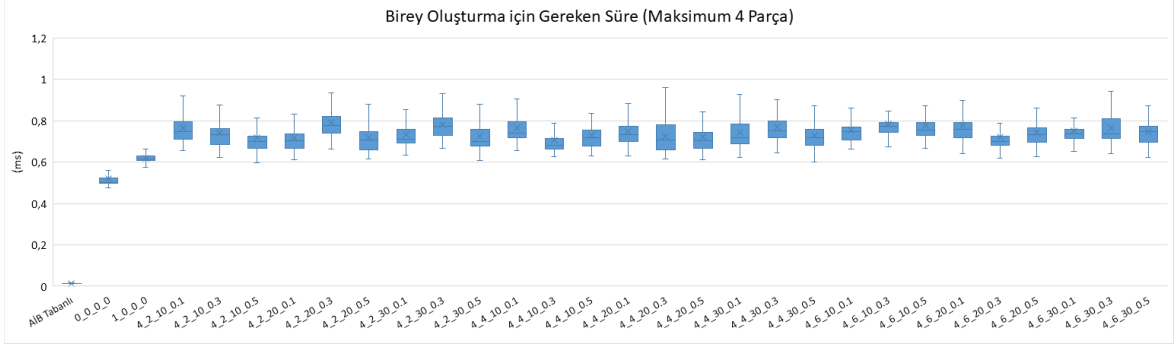
Öngörülebileceği üzere yeni birey oluşturma sürecinde çok çok az hesaplama gerçekleştiren RVO yönteminin $0.003\ ms$ değeri ile açık ara en düşük ortalama birey oluşturma süresine sahip olduğu Şekil 6.38’de verilen kutu grafiğinde de görülmektedir. Diğer yöntemler ile gerçekleştirilen benzetimlerde birey oluşturmak için harcanan ortalama süreleri gösteren kutu grafikleri ise Şekil 6.39’da sunulmaktadır. Buna ek olarak, sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, sırasıyla Şekil 6.40, Şekil 6.41 ve Şekil 6.42’de verilmiştir. Bu sonuçlara göre, AİB tabanlı TYPÇY metodunun birey oluşturmak için ortalama $0.014\ ms$ süreye ihtiyaç duyduğu belirlenmiştir (Şekil 6.38). Bu değer, her ne kadar RVO yönteminden sonraki en düşük birey oluşturma süresine karşılık geliyor olsa da AİB tabanlı TYPÇY metodu birey oluşturma işlemleri için RVO yöntemine göre yaklaşık 4 kat daha fazla zaman gerektirmektedir. GİB tabanlı TYPÇY yönteminin birey oluşturmak için harcadığı ortalama süre $0.519\ ms$ kadardır. Bu değer, mikro saniyeler mertebesinde birey oluşturabilen RVO yönteminden çok yüksek olmakla birlikte AİB tabanlı TYPÇY metodunun ihtiyaç duyduğu birey oluşturma süresinin yaklaşık 36 katıdır. Buna karşın, ÇYPCY yönteminin sadece TYPÇY’lere sahip bireyler oluşturmak için ihtiyaç duyduğu ortalama süre $0.618\ ms$ olarak ölçülmüştür. Yine bu değer de RVO yönteminden çok yüksek olmakla birlikte AİB tabanlı TYPÇY metodunun ihtiyaç duyduğu birey oluşturma süresinin yaklaşık 43 katıdır.



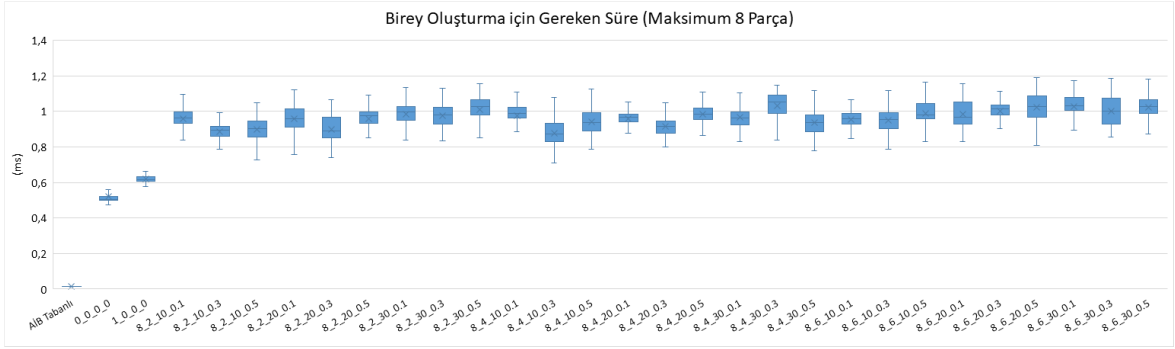
Şekil 6.38 RVO ve AİB tabanlı TYPÇY yöntemleriyle gerçekleştirilen benzetimler için ortalama birey oluşturma süreleri ($r = 0.5 m$)



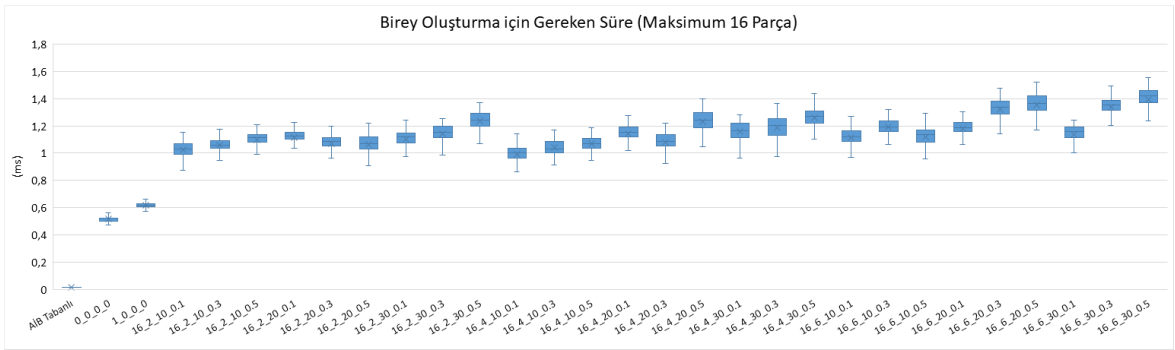
Şekil 6.39 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)



Şekil 6.40 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)



Şekil 6.41 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)

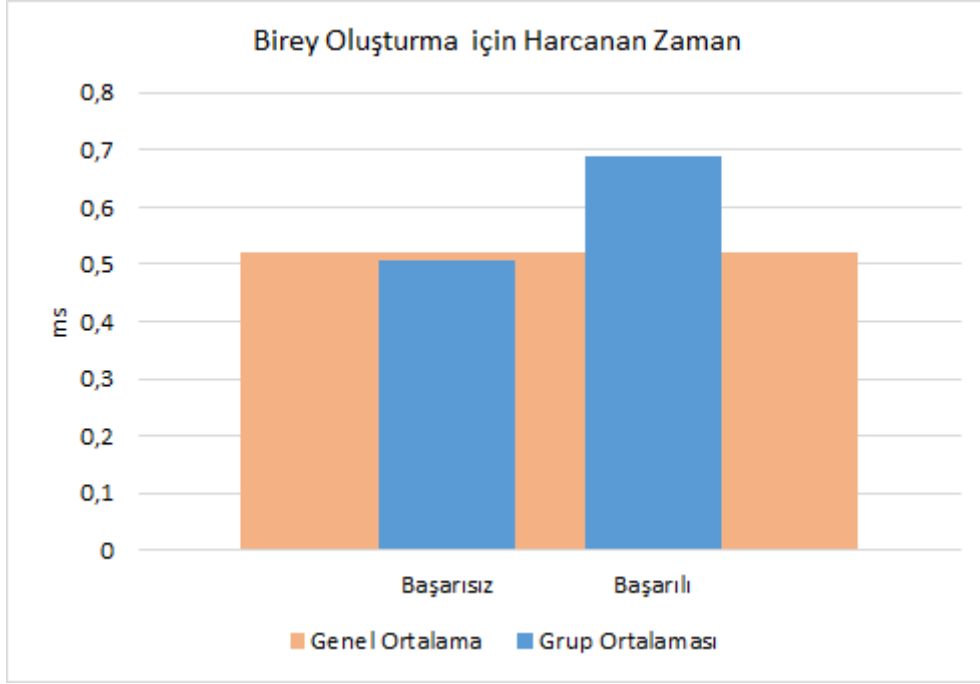


Şekil 6.42 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.5 m$)

Daha önce de belirtildiği üzere RVO yöntemi, birey oluşturmak üzere çok az miktarda hesaplama gerçekleştirdiği için birey oluşturma maliyeti bakımından TYPÇY'ler kullanan bireyler üreten tüm diğer yöntemlerden üstün gelmektedir. AİB tabanlı TYPÇY metodunun GİB tabanlı TYPÇY ve sadece TYPÇY'ler kullanan ÇYPÇY yöntemlerinden daha düşük birey oluşturma maliyetine sahip olmasının ana nedeni, AİB ile GİB arasında hiçbir yönde veri transferi gerektirmemesidir. GİB tabanlı TYPÇY ve ÇYPÇY metotlarının her ikisi de özellikle daha yavaş bir aktarım olduğu bilinen GİB belleğinden AİB belleğine veri transferi gerçekleştirdiği için daha yüksek birey oluşturma sürelerine ihtiyaç duymaktadır. Yine de GİB tabanlı TYPÇY ve sadece TYPÇY'ler kullanan ÇYPÇY yöntemlerinin her ikisinin de yarıçapı 0.5 m olan bir birey oluşturmak için milisaniyenin daha altında sürelere ihtiyaç duymakla birlikte geçerli olup olmadığı önceden kestirilemeyen tamamen rastgele çözümler deneyen AİB tabanlı TYPÇY metoduna kıyasla ortalama birey sayısında yaklaşık %34'e kadar ve maksimum birey sayısında yaklaşık %48'e kadar daha fazla birey oluşturabildiğini belirtmekte fayda vardır.

Diğer yandan, benzetimlerde ortalama ve maksimum birey sayıları birbirine çok yakın olmasına rağmen, ÇYPÇY yöntemi, sadece TYPÇY'lere sahip bireyler oluşturmak üzere GİB tabanlı TYPÇY yönteminden yaklaşık %19 daha fazla süreye ihtiyaç duymaktadır. Bu durumun temel nedeni, GİB tabanlı TYPÇY yönteminde kullanılan GPGPU geçişlerinden ikincisinin her durumda gerçekleştirilmemesidir. Hatırlanacağı üzere GİB tabanlı TYPÇY yönteminin ilk GPGPU geçişinin sonunda bir atomik sayaç, komşu beyaz piksellerin birbirinden farklı kümelerinin sayısını tutmaktadır. Atomik sayacın değerinin sıfırdan büyük olması olası çözümlere karşılık gelen beyaz piksellerin var olduğuna, dolayısıyla sürdürülmekte olan birey oluşturma denemesi sonucunda yeni bir birey oluşturulabileceğine işaret eder. “Başarılı” birey oluşturma denemesi olarak adlandırılan bu senaryonun tam aksine atomik sayacın değerinin sıfır olması durumunda, olası çözüm olarak kullanılacak hiçbir beyaz piksel bulunmadığı için yeni bir birey oluşturulamayacak ve deneme “başarısız” olarak kabul edilecektir. GİB tabanlı TYPÇY yönteminde, ilk GPGPU geçişinin tamamlanmasının hemen sonrasında atomik sayacın değerine bakılarak birey oluşturma denemesinin başarılı mı başarısız mı olacağına dair fikir sahibi olunmaktadır. Eğer denemenin başarılı olacağı öngörülüyorsa sıradaki adım olan ikinci GPGPU geçişi işleme alınarak yeni birey için çarpışmasız yol belirleme sürecine devam edilirken, tam tersine eğer denemenin başarısız olacağı öngörülüyorsa söz konusu bu ikinci geçişin ve sonrasındaki diğer bütün adımların işletimi iptal edilerek deneme sonlandırılmaktadır. GİB tabanlı

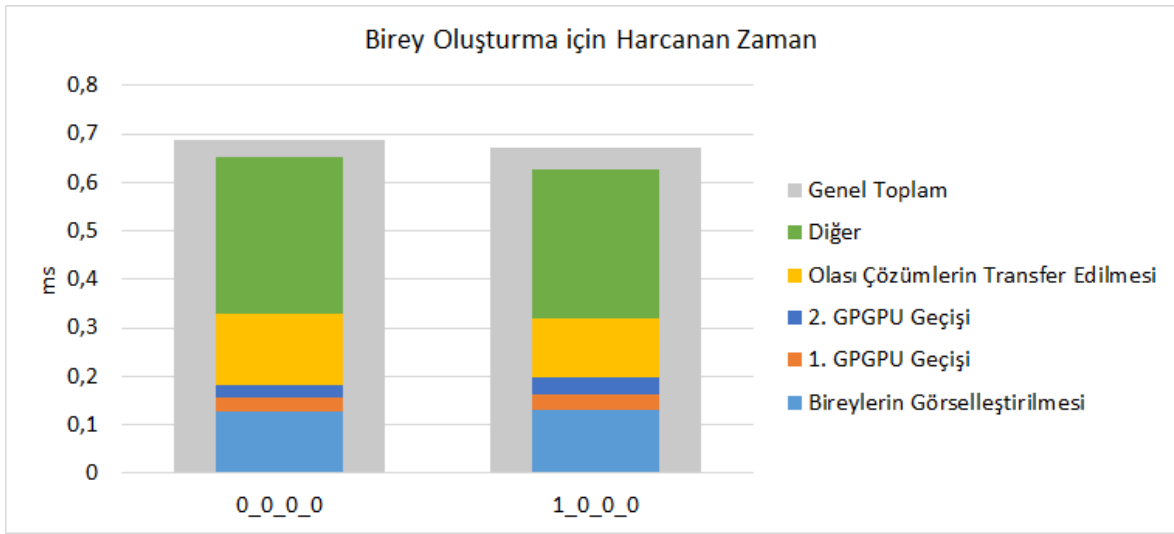
TYPÇY yönteminin, işletim kaynaklarını korumaya yönelik bu yaklaşımı neticesinde başarılı birey oluşturma denemeleri, başarısız birey oluşturma denemelerine göre hatırı sayılır miktarda daha fazla zaman gerektirmektedir. Şekil 6.43'te başarılı (0.687 *ms*) ve başarısız (0.508 *ms*) birey oluşturma denemelerinin gerektirdiği sürelerinin ayrı ayrı ortalamalarının yanı sıra referans oluşturması açısından GİB tabanlı TYPÇY yönteminin her iki grubu da kapsayan ortalama birey oluşturma süresi (0.519 *ms*) de verilmiştir.



Şekil 6.43 GİB tabanlı TYPÇY yönteminde başarılı ve başarısız birey oluşturma denemeleri için harcanan ortalama süreler ($r = 0.5 m$)

ÇYPÇY yönteminin ÇYPÇY'lere sahip bireyler oluşturmak için ihtiyaç duyduğu ortalama süre, kullanılan parametre değerlerinin kombinasyonuna bağlı olarak [0.706, 1.394] *ms* aralığında değişiklik göstermektedir. Bu sonuçlar göz önünde bulundurulduğunda, 0.5 *m* yarıçapa ve ÇYPÇY'lere sahip bireyler üreten benzetimlerde, ortalama birey sayılarında %53'e, maksimum birey sayılarında %47'ye varan artışlara ulaşabilmek adına ihtiyaç duyulan birey oluşturma sürelerinin, GİB tabanlı TYPÇY yöntemine göre %36-168, ÇYPÇY yönteminin sadece TYPÇY'lere izin verilen benzetimine göre ise %14-125 artış gösterdiği görülmektedir. Bu durumun esas nedeni, geliştirilen GİB tabanlı çarpışmasız yol belirleme yöntemlerinde birey oluşturma maliyetinin büyük bölümünü çarpışmasız yörünge parçalarının belirlenmesi işlemlerinin oluşturmasıdır. Şekil 6.44'te GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere çarpışmasız

yörünge parçalarını belirlemek için gerçekleştirilen farklı işlemlerin harcadığı sürelerin yanı sıra birey oluşturma denemeleri (GİB tabanlı TYPÇY yöntemi için yalnızca başarılı denemeler) için gereken toplam sürelerin ortalaması da verilmiştir. ÇYPÇY'ler kullanan benzetimlerde bu yörünge parçalarının sayısındaki artışa bağlı olarak bireylere atanacak çarpışmasız yolların belirlenmesi için gereken süre de doğru orantılı bir biçimde artmaktadır. Ayrıca ÇYPÇY yönteminde kullanılan parametrelerin daha geniş çözüm uzayları sunan kombinasyonlarında olası çözümlerin sayısı arttığı için tüm bu olası çözümlerin işlenmesi de zaman maliyetinde küçük de olsa bir miktar artışı beraberinde getirmektedir.



Şekil 6.44 GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere yörünge parçası belirleme sürecinin farklı alt işlemleri tarafından harcanan süreler ($r = 0.5 m$)

Bunun yanı sıra, ÇYPÇY yönteminin ÇYPÇY'ler kullanan benzetimlerinden elde edilen veriler üzerinde dört yönlü bir “varyans analizi” (*Analysis of Variance - ANOVA*) gerçekleştirilmiştir. Bu amaçla birey oluşturma için harcanan ortalama süre bağımlı değişken olarak kullanılırken *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametreleri bağımsız değişkenler olarak ele alınmıştır. Çoklu karşılaştırma analizleri için Tukey HSD testi kullanılırken tüm bu istatistiksel analizlerde güven seviyesi %95 olarak ayarlanmıştır. Analiz sonuçları incelendiğinde *max_num_of_segments* ($F(2, 72) = 383.609$, $p \approx 0$), *margin_coef* ($F(2, 72) = 11.663$, $p \approx 0$) ve *delta_angle* ($F(2, 72) = 9.875$, $p \approx 0$) parametrelerinin ortalama birey oluşturma süresi üzerinde ana etki sahibi olduğu belirlenmiştir (Çizelge 6.9). Bu bağımsız değişkenlerin etki büyüklükleri sırasıyla; *max_num_of_segments* için $\eta^2 = 0.866$, *margin_coef* için $\eta^2 = 0.026$ ve *delta_angle* için η^2

= 0.022 şeklindedir (Çizelge 6.9). Çoklu karşılaştırma testlerinin sonuçlarına göre ortalama birey oluşturma süresi maliyeti, izin verilen maksimum yörünge parçası sayısı ile doğru orantılıdır (tüm durumlar için $p \approx 0$). Buna ek olarak, yörünge parçalarının bitim noktalarının geri çekilme payı olarak $6 * r$ tercih edilmesi, $4 * r$ veya $2 * r$ seçilen benzetimlere kıyasla ortalama birey oluşturma sürelerinde artışa neden olmaktadır ($4 * r$ için $p < 0.001$ ve $2 * r$ için $p \approx 0$). Son olarak, yörünge parçalarının birleşim noktalarında yürüme doğrultusunda en fazla 20° ($p < 0.036$) veya 30° ($p \approx 0$) açı değişikliklerine izin vermek, en fazla 10° doğrultu değişikliklerine izin verilen benzetimlere göre birey oluşturmak için harcanan ortalama sürede artışa neden olmaktadır.

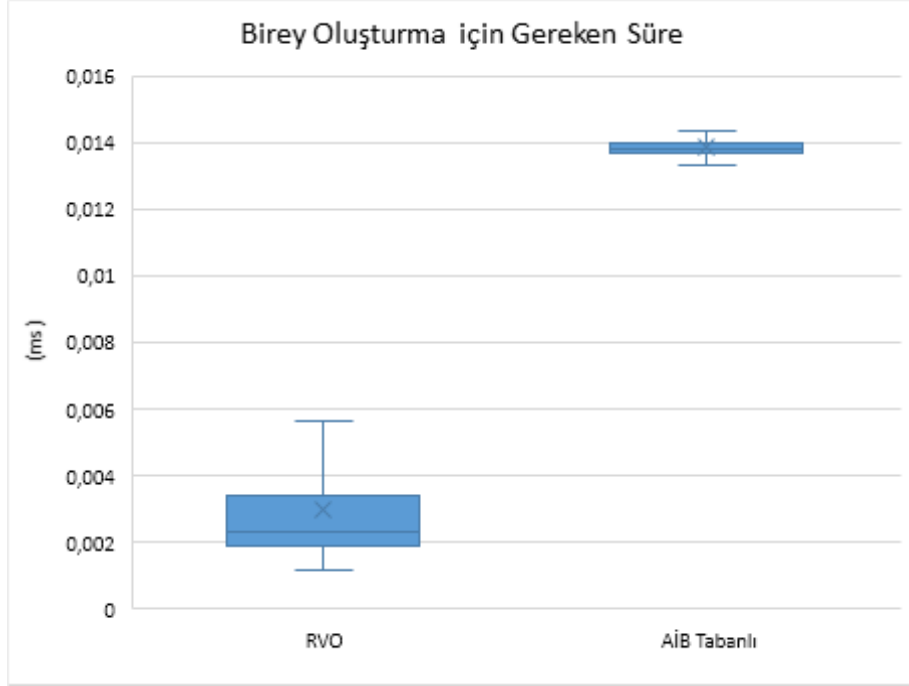
Çizelge 6.9 ÇYPCY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey oluşturma süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 m$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Birey oluşturma süresi	max_num_of_segments	2	383.609	0.000	0.866
	margin_coef	2	11.663	0.000	0.026
	delta_angle	2	9.875	0.000	0.022
	delta_speed	2	1.609	0.207	0.004
	Hata	72			

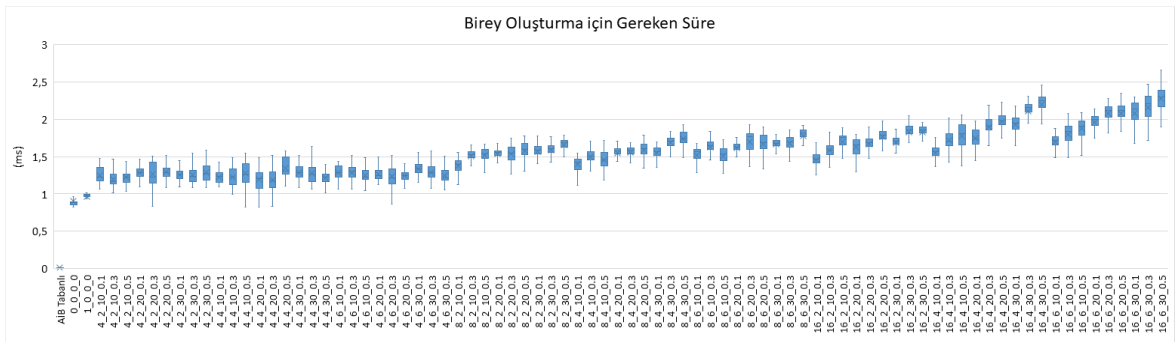
6.3.1.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar

Daha önce de belirtildiği üzere RVO yöntemi, birey oluşturma sürecinde çok çok az hesaplama gerçekleştirdiği için Şekil 6.45'te verilen kutu grafiğinde de görülebileceği gibi $0.003 ms$ değeri ile açık ara en düşük ortalama birey oluşturma süresine sahiptir. Geliştirilen tüm yöntemler ile gerçekleştirilen benzetimlerde birey oluşturmak için harcanan ortalama süreleri gösteren kutu grafikleri ise Şekil 6.46'da sunulmaktadır. Ayrıca sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, sırasıyla Şekil 6.47, Şekil 6.48 ve Şekil 6.49'da verilmiştir. Elde edilen sonuçlar göz önünde bulundurulduğunda, AİB tabanlı TYPÇY metodunun birey oluşturmak için ortalama $0.014 ms$ süreye ihtiyaç duyduğu tespit edilmiştir (Şekil 6.45). Bu değer, RVO yönteminden sonraki en düşük birey oluşturma süresine karşılık gelmekle birlikte AİB tabanlı TYPÇY metodu, birey oluşturma işlemleri için RVO yöntemine göre yaklaşık 4 kat daha fazla zamana ihtiyaç duymaktadır. GİB tabanlı TYPÇY yönteminin birey oluşturmak için harcadığı ortalama süre $0.902 ms$ kadardır. Bu değer, mikro saniyeler mertebesinde birey oluşturabilen RVO yönteminden çok yüksek olmanın

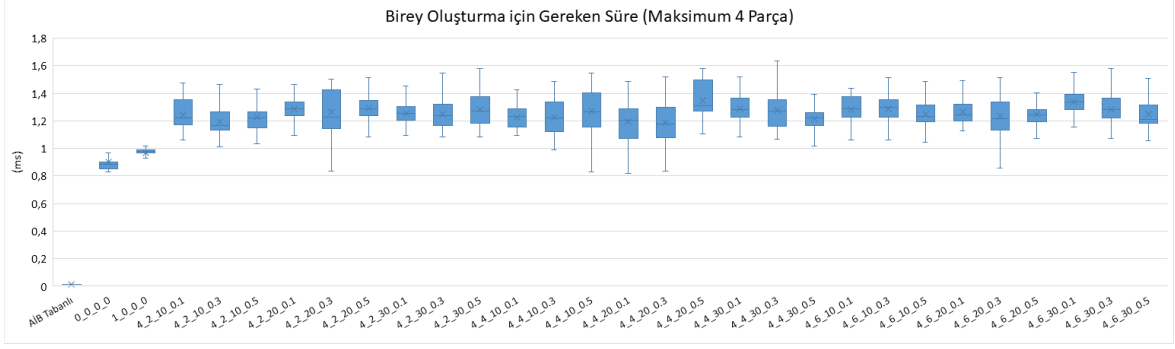
yanı sıra AİB tabanlı TYPÇY metodunun ihtiyaç duyduğu birey oluşturma süresinin yaklaşık 65 katıdır. Diğer taraftan, ÇYPCY yönteminin sadece TYPÇY'lere sahip bireyler oluşturmak için ihtiyaç duyduğu ortalama süre $0.968\ ms$ olarak ölçülmüştür. Benzer şekilde, bu değer de RVO yönteminden çok yüksek olmakla birlikte AİB tabanlı TYPÇY metodunun ihtiyaç duyduğu birey oluşturma süresinin yaklaşık 70 katıdır.



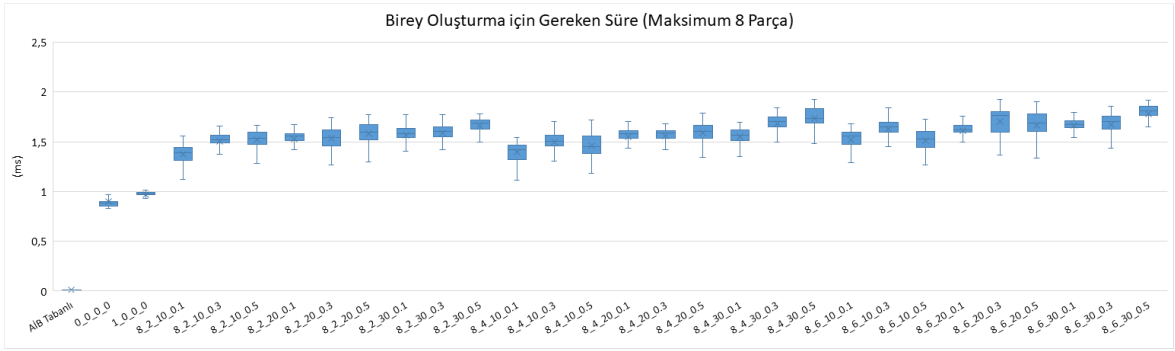
Şekil 6.45 RVO ve AİB tabanlı TYPÇY yöntemleriyle gerçekleştirilen benzetimler için ortalama birey oluşturma süreleri ($r = 0.3\ m$)



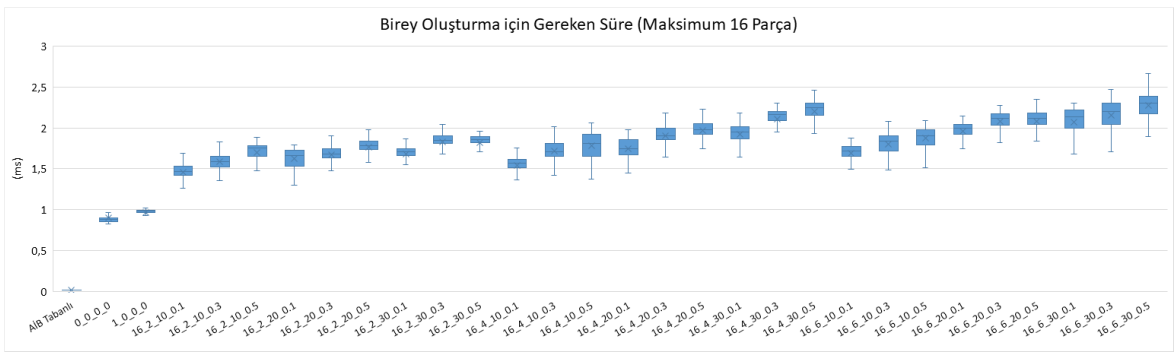
Şekil 6.46 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3\ m$)



Şekil 6.47 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)



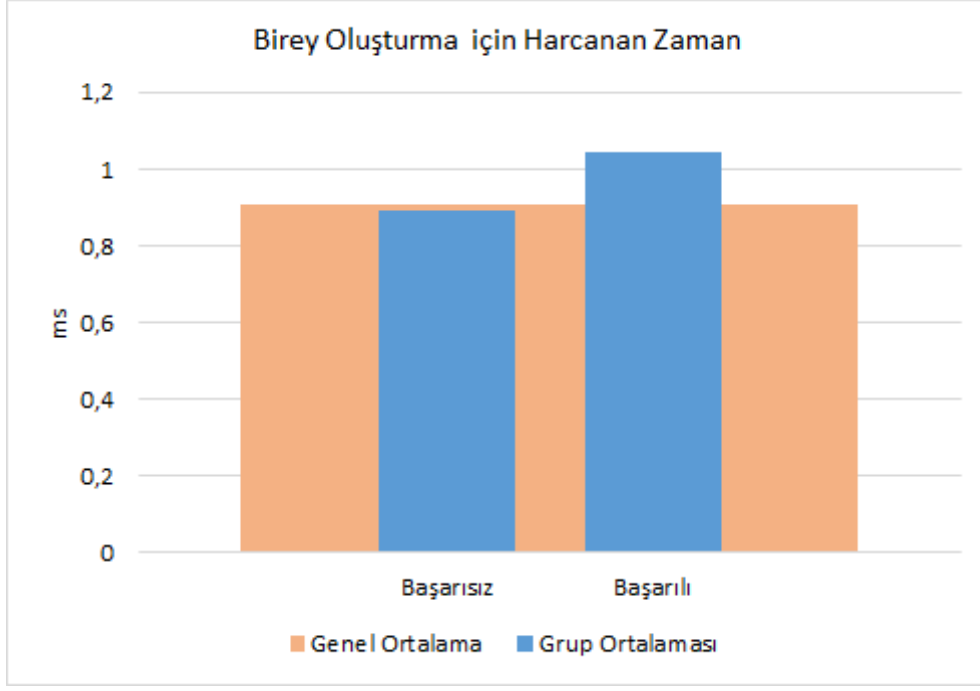
Şekil 6.48 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)



Şekil 6.49 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için ortalama birey oluşturma süreleri ($r = 0.3 m$)

Öngörülebileceği üzere RVO yöntemi, birey oluşturmak üzere çok az miktarda hesaplama gerçekleştirdiği için birey oluşturma maliyeti bakımından TYPÇY'ler kullanan bireyler üreten tüm diğer yöntemlerden daha avantajlıdır. AİB tabanlı TYPÇY metodu, AİB ile GİB arasında hiçbir yönde veri transferine ihtiyaç duymadığı için GİB tabanlı TYPÇY ve sadece TYPÇY'ler kullanan ÇYPÇY yöntemlerinden daha düşük birey oluşturma maliyetine sahiptir. GİB tabanlı TYPÇY ve ÇYPÇY metotlarının daha yüksek birey oluşturma süreleri gerektirmelerinin temel nedeni, her ikisinin de özellikle daha yavaş bir aktarım olduğu bilinen GİB belleğinden AİB belleğine veri transferi gerçekleştirmesidir. Buna rağmen, GİB tabanlı TYPÇY ve sadece TYPÇY'ler kullanan ÇYPÇY yöntemlerinin her ikisi de yarıçapı 0.3 m olan bir birey oluşturmak için milisaniyenin altında sürelere ihtiyaç duymakla birlikte tamamen rastgele çözümler deneyerek büyük oranda başarısızlıkla sonuçlanan AİB tabanlı TYPÇY metoduna kıyasla ortalama birey sayısında yaklaşık %80'e kadar ve maksimum birey sayısında yaklaşık %99'a kadar daha fazla birey oluşturabilmektedir.

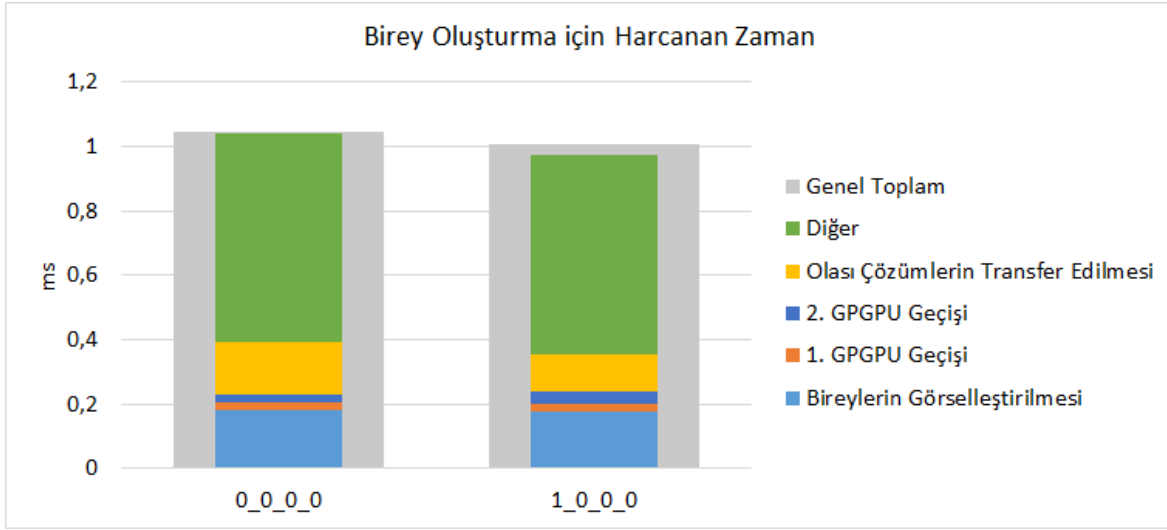
GİB tabanlı TYPÇY ve sadece TYPÇY'ler kullanan ÇYPÇY yöntemleriyle gerçekleştirilen benzetimlerde ortalama ve maksimum birey sayıları birbirine çok yakındır, ancak ÇYPÇY yöntemi sadece TYPÇY'lere sahip bireyler oluşturmak üzere GİB tabanlı TYPÇY yönteminden yaklaşık %7 daha fazla süreye ihtiyaç duymaktadır. Daha önce de açıklandığı üzere bu farklılığın ana nedeni, GİB tabanlı TYPÇY yönteminde kullanılan ilk GPGPU geçişinde hiçbir beyaz piksel bulunmadığı için işlem yapılamaması durumunda, GPGPU geçişlerinden ikincisinin gerçekleştirilmemesidir. GİB tabanlı TYPÇY yönteminin birey oluşturma maliyetini azaltmayı hedefleyen bu yaklaşımı sayesinde başarılı birey oluşturma denemeleri, başarısız birey oluşturma denemelerine göre fark oluşturacak miktarda daha fazla zamana ihtiyaç duymaktadır. Şekil 6.50'de başarılı (1.046 ms) ve başarısız (0.891 ms) birey oluşturma denemelerinin ihtiyaç duyduğu sürelerin ayrı ayrı ortalamalarının yanı sıra referans oluşturması açısından GİB tabanlı TYPÇY yönteminin her iki grubu da kapsayan ortalama birey oluşturma süresi (0.902 ms) de verilmiştir.



Şekil 6.50 GİB tabanlı TYPÇY yönteminde başarılı ve başarısız birey oluşturma denemeleri için harcanan ortalama süreler ($r = 0.3 m$)

ÇYPÇY yönteminin ÇYPÇY'lere sahip bireyler oluşturmak için ihtiyaç duyduğu ortalama süre, kullanılan parametre değerlerinin kombinasyonuna bağlı olarak değişmekle birlikte [1.188, 2.274] ms aralığında yer almaktadır. Bu sonuçlara göre, 0.3 m yarıçapa ve ÇYPÇY'lere sahip bireyler üreten benzetimlerde, ortalama birey sayılarında %58'e, maksimum birey sayılarında %50'ye varan artışlar sağlayabilmek için gereken birey oluşturma sürelerinin, GİB tabanlı TYPÇY yöntemine göre %32-152, ÇYPÇY yönteminin sadece TYPÇY'ler kullanan benzetimine göre ise %23-135 artış gösterdiği belirlenmiştir. Önceki bölümde de değinildiği üzere, geliştirilen GİB tabanlı çarpışmasız yol belirleme yöntemlerinde birey oluşturma maliyetinin büyük bölümünü çarpışmasız yörünge parçalarının belirlenmesi işlemleri oluşturmaktadır. Şekil 6.51'de GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere çarpışmasız yörünge parçalarını belirlemek için gerçekleştirilen farklı işlemlerin harcadığı sürelerle ek olarak birey oluşturma denemeleri (GİB tabanlı TYPÇY yöntemi için yalnızca başarılı denemeler) için gereken toplam sürelerin ortalaması da verilmiştir. ÇYPÇY'ler kullanan benzetimlerde yörünge parçalarının sayısındaki artışa paralel olarak bireylere atanacak çarpışmasız yolların belirlenmesi için gereken süre de artmaktadır. Buna ek olarak, ÇYPÇY yönteminde kullanılan parametrelerin daha geniş çözüm uzayları sunan kombinasyonlarında

olası çözümlerin sayısı artmakta ve tüm bu olası çözümlerin işlenmesi de zaman maliyetinde az da olsa bir miktar artışa neden olmaktadır.



Şekil 6.51 GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinde yalnızca TYPÇY'lere sahip bireyler oluşturmak üzere yörünge parçası belirleme sürecinin farklı alt işlemleri tarafından harcanan süreler ($r = 0.3 m$)

Tüm bunlara ek olarak, ÇYPCY yönteminin ÇYPCY'ler kullanan benzetimlerinden elde edilen veriler üzerinde dört yönlü bir ANOVA gerçekleştirilmiştir. Bu amaçla birey oluşturma için harcanan ortalama süre bağımlı değişken olarak kullanılırken *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametreleri bağımsız değişkenler olarak ele alınmıştır. Çoklu karşılaştırma analizleri için Tukey HSD testi kullanılırken tüm bu istatistiksel analizlerde güven seviyesi %95 olarak ayarlanmıştır. Analiz sonuçları değerlendirildiğinde *max_num_of_segments* ($F(2, 72) = 260.193, p \approx 0$), *margin_coef* ($F(2, 72) = 15.335, p \approx 0$), *delta_angle* ($F(2, 72) = 21.742, p \approx 0$) ve *delta_speed* ($F(2, 72) = 6.333, p < 0.003$) parametrelerinin ortalama birey oluşturma süresi üzerinde ana etki sahibi olduğu görülmüştür (Çizelge 6.10). Bu bağımsız değişkenlerin etki büyüklükleri sırasıyla; *max_num_of_segments* için $\eta^2 = 0.766$, *delta_angle* için $\eta^2 = 0.064$, *margin_coef* için $\eta^2 = 0.045$ ve *delta_speed* için $\eta^2 = 0.019$ olarak belirlenmiştir (Çizelge 6.10). Çoklu karşılaştırma testlerinin sonuçları göz önünde bulundurulduğunda, ortalama birey oluşturma süresi maliyeti, izin verilen maksimum yörünge parçası sayısı ile doğru orantılıdır (tüm durumlar için $p \approx 0$). Ayrıca yörünge parçalarının bitim noktalarının geri çekilme payının artırılmasına paralel olarak ortalama birey oluşturma süresi maliyeti de artmaktadır ($2 * r$ yerine $4 * r$ için $p < 0.028$, $4 * r$ yerine $6 * r$ için $p < 0.013$ ve $2 * r$ yerine

$6 * r$ için $p \approx 0$). Aynı şekilde, yörünge parçalarının birleşim noktalarında izin verilen maksimum doğrultu değişikliği açısının artırılmasıyla birlikte ortalama birey oluşturma süresi maliyeti de artmaktadır (10° yerine 20° için $p < 0.001$, 20° yerine 30° için $p < 0.016$ ve 10° yerine 30° için $p \approx 0$). Son olarak, yörünge parçalarının birleşim noktalarında yürüme hızında en fazla $0.5 m/s$ hız değişikliklerine izin vermek, en fazla $0.1 m/s$ hız değişikliklerine izin verilen benzetimlere göre birey oluşturmak için harcanan ortalama sürede artışa neden olmaktadır ($p < 0.002$).

Çizelge 6.10 ÇYPCY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey oluşturma süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 m$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Birey oluşturma süresi	max_num_of_segments	2	260.193	0.000	0.766
	delta_angle	2	21.742	0.000	0.064
	margin_coef	2	15.335	0.000	0.045
	delta_speed	2	6.333	0.003	0.019
	Hata	72			

Önceki bölümlerde belirtildiği üzere aynı benzetim alanı üzerinde daha küçük yarıçapa sahip çok daha fazla birey oluşturmak mümkün olabilmektedir. Ancak benzetim alanı içerisindeki birey yoğunluğu arttıkça özellikle geliştirilen GİB tabanlı yöntemlerin birey oluşturma maliyetleri de bununla birlikte artmaktadır. Elde edilen sonuçlar göz önünde bulundurulduğunda, GİB tabanlı TYPÇY yönteminin birey oluşturma maliyetindeki bu artış %74 iken ÇYPCY yönteminin yalnızca TYPÇY’ler kullanan benzetimlerinde %57 civarındadır. Benzer biçimde, ÇYPCY yönteminin ÇYPCY’ler kullanan benzetimlerinde ise parametrelerin farklı kombinasyonlarına bağlı olarak söz konusu bu artışın miktarı %43-87 aralığında değişmektedir. RVO yönteminde yarıçapı $0.3 m$ olan bireyler kullanmanın yarıçapı $0.5 m$ olan bireyler kullanmaya kıyasla birey oluşturma maliyetinde yol açtığı artış %1 kadardır, ancak %95 güven seviyesinde gerçekleştirilen “bağımsız gruplar arası t testi” (*Independent Samples t-test*) sonucunda bu değişimin istatistiksel açıdan anlamlı olmadığı görülmüştür ($t(8) = 0.113$, $p = 0.913$). Diğer taraftan, AİB tabanlı TYPÇY yönteminde yarıçapı $0.5 m$ olan bireyler yerine yarıçapı $0.3 m$ olan bireyler kullanılması, birey oluşturma maliyetini yaklaşık %4 azaltmaktadır. Bu duruma benzetim alanı üzerinde daha az alan işgal eden bireylerin nispeten daha az sayıda ızgara hücresi ile etkileşime girmesi sonucu gerçekleştirilen hesaplamalardaki azalmanın neden olduğu düşünülmektedir.

6.3.2. Birey Gezinimi Maliyetlerinin Karşılaştırılması

Gezinim işlemlerini kapsayan diğer gruptaki hesaplamalardan söz etmek gerekirse, geliştirilen gezinim yöntemleri, her bir benzetim zaman adımında bireylerin hız vektörlerine ve geçen süreye bağlı olarak basit bir biçimde yer değiştirmelerini hesaplayıp sadece bireylerin pozisyonlarını güncellediği için hemen hemen hiç maliyetli değildir. Her ne kadar bu gezinim yöntemleri ihmal edilebilir bir gezinim maliyetine sahip olduğundan geliştirilen bu üç yöntemin gezinim maliyetlerini karşılaştırmak gereksiz olsa da bu yöntemlerin maliyetlerinin alanda var olan gezinim yöntemleri ile kıyaslanması gereklidir. Bu amaçla bir kenarının uzunluğu 128 m olan kare biçiminde bir benzetim alanı üzerinde tüm yöntemler için ayrı ayrı 5 dakikalık benzetimler çalıştırılmıştır. Başlangıçta boş olan benzetim alanını olabildiğince fazla birey oluşturarak doldurmaya çalışan bu benzetimler, çevrimdışı olarak gerçekleştirilmiş ve zaman adımlarının arasında 1 ms süre olacak biçimde ayarlanmıştır. Ayrıca bu benzetimlerde her bir zaman adımında birey oluşturma için ayrılan süre sınırı kaldırılmış ancak her bir zaman adımı içerisinde yalnızca tek bir birey oluşturma denemesi yapılmasına izin verilmiştir. Benzetim boyunca her bir zaman adımında benzetimde yer alan tüm bireylerin gezinimini sağlamak için harcanan toplam süre ölçülerek birey sayısına bölünmek suretiyle gezinim için birey başına harcanan ortalama süre değeri elde edilmiştir. Benzetim sonunda her bir zaman adımında elde edilen bu değerlerin ortalaması alınarak benzetim genelinde gezinim için birey başına harcanan ortalama süre hesaplanmıştır. ÇYPCY yönteminde *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametreleri için önceki deneylerle aynı değerler kullanılmış ve benzer biçimde bu parametre değerlerinin farklı her bir kombinasyonu için ayrı bir benzetim çalıştırılmıştır. Tüm deneyler birey yarıçapının 0.5 m ve 0.3 m olmak üzere iki farklı değeri için iki farklı grup olarak gerçekleştirilmiştir.

Bir önceki bölümde yapılan birey oluşturma maliyeti değerlendirmesinde de bahsedildiği üzere RVO yönteminde birey oluşturma sürecinde yapılan basit hesaplamalar, birey sayısı üst sınırı için önemli bir engel teşkil etmediğinden RVO yönteminde maksimum birey sayısı doygunluğu çok daha yüksektir. Bu nedenle RVO yöntemi, benzetim süresince rakiplerine göre daha fazla birey oluşturacak ve her bir bireyin yakın çevresinde artan sayıda başka bireyi göz önünde bulundurarak gezinimini sağlamak için uğraşırken çok daha fazla hesaplama yapmak suretiyle dezavantajlı duruma düşecektir. Bunun önüne geçmek adına bir önceki bölümde olduğu gibi RVO yönteminde birey sayısı üst sınırı, birey yarıçapının 0.5 m olduğu deney grubu için 1600, birey yarıçapının 0.3 m olduğu deney grubu için ise 2900

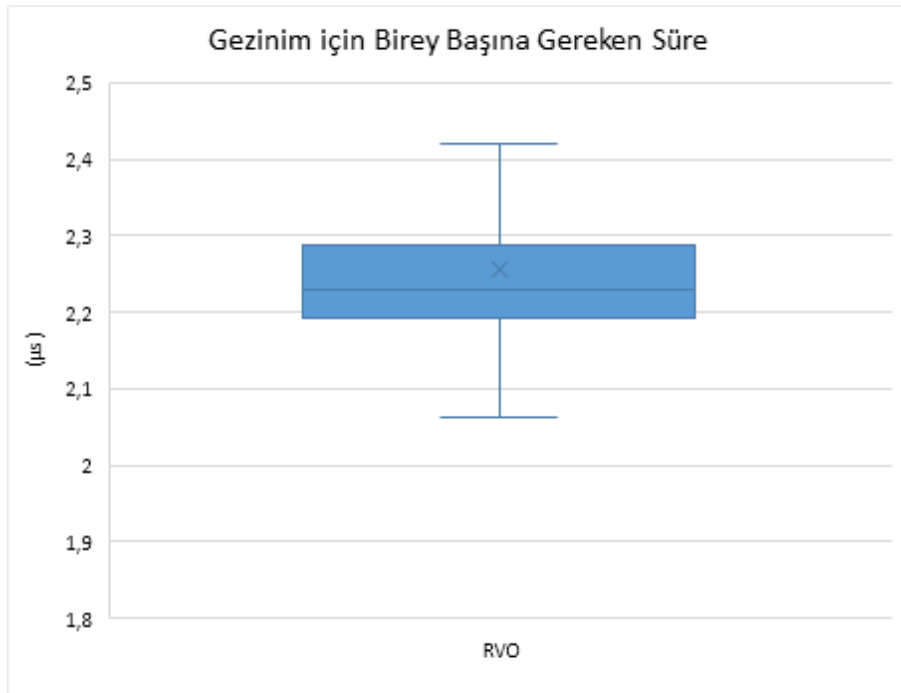
olarak belirlenmiş ve RVO yönteminin birey sayısı bu iki değeri geçmeyecek şekilde birey oluşturarak sabit birey yoğunluklarına sahip olması sağlanmıştır. Daha önce de değinildiği gibi bu değerler, ÇYPCY yönteminin ulaşabildiği maksimum ortalama birey sayısı değerlerini geçmeyecek biçimde seçilmiştir.

6.3.2.1. Birey Yarıçapının 0.5 m Değeri için Sonuçlar

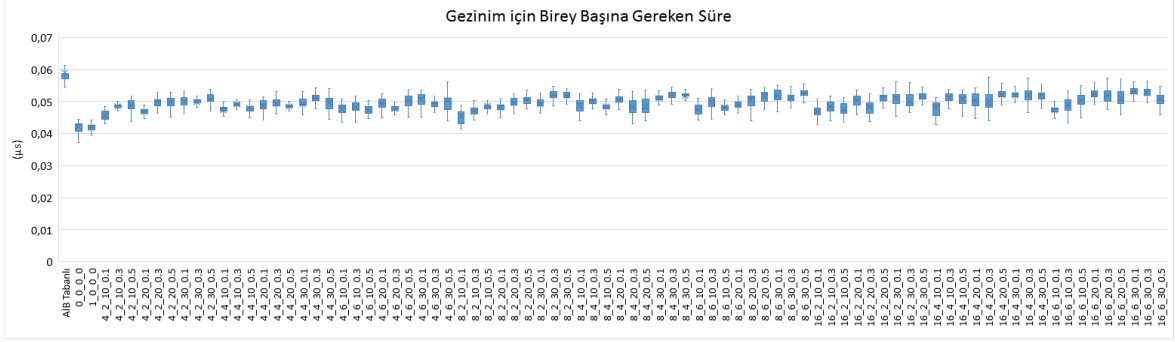
RVO yöntemi ile gerçekleştirilen benzetimde gezinim için birey başına harcanan ortalama sürenin kutu grafiği Şekil 6.52’de verilmiştir. Tahmin edilebileceği üzere her bir bireyin gezinimini sağlamak adına yakın çevresini kontrol ederek burada yer alan tüm diğer bireyleri göz önünde bulundurmamak suretiyle yeni bir çarpışmasız hız vektörü bulmaya çalışan RVO yöntemi $2.256 \mu s$ değeri ile gezinim için birey başına harcanan süre konusunda en yüksek maliyete sahiptir. Diğer yöntemler ile gerçekleştirilen benzetimlerde gezinim için birey başına harcanan ortalama süreleri gösteren kutu grafikleri ise Şekil 6.53’te sunulmuştur. Bununla birlikte sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, sırasıyla Şekil 6.54, Şekil 6.55 ve Şekil 6.56’da verilmiştir. Elde edilen bu sonuçlara göre, AİB tabanlı TYPÇY metodunun gezinim için birey başına ortalama $0.059 \mu s$ süreye ihtiyaç duyduğu belirlenmiştir. Bu değer, RVO yönteminden sonraki en yüksek birey başına gezinim süresine karşılık gelmekle birlikte RVO yöntemi, birey başına gezinim işlemleri için AİB tabanlı TYPÇY metoduna göre yaklaşık 37 kat daha fazla zaman gerektirmektedir. AİB tabanlı TYPÇY metodunun RVO yönteminden daha düşük gezinim maliyetine sahip olmasının en temel nedeni, “yönlendirmesiz” olması yani her bir birey için yüksek maliyetli çarpışma tespiti ve çarpışmadan kaçınma işlemleri gerçekleştirilmemesidir.

GİB tabanlı TYPÇY yönteminin gezinim için birey başına harcadığı ortalama süre $0.042 \mu s$ kadardır. Birey başına en yüksek gezinim maliyetine sahip RVO yönteminin birey başına ortalama gezinim süresi, bu değer yaklaşık 53 katı olmakla birlikte AİB tabanlı TYPÇY metodu, gezinim için birey başına bu değerden yaklaşık %40 daha fazla zamana ihtiyaç duymaktadır. Buna karşın, ÇYPCY yönteminin sadece TYPÇY’lere sahip bireylerin gezinimi için birey başına ihtiyaç duyduğu ortalama süre yaklaşık $0.042 \mu s$ olarak ölçülmüştür. Yine RVO yönteminin birey başına ortalama gezinim süresi, bu değer yaklaşık 53 katı olmakla birlikte AİB tabanlı TYPÇY metodu, gezinim için birey başına bu değerden yaklaşık %41 daha fazla zamana ihtiyaç duymaktadır. Her biri yönlendirmesiz olmasına karşın GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin AİB tabanlı TYPÇY metodundan daha düşük gezinim maliyetine sahip olmasının en önemli nedeni, AİB tabanlı

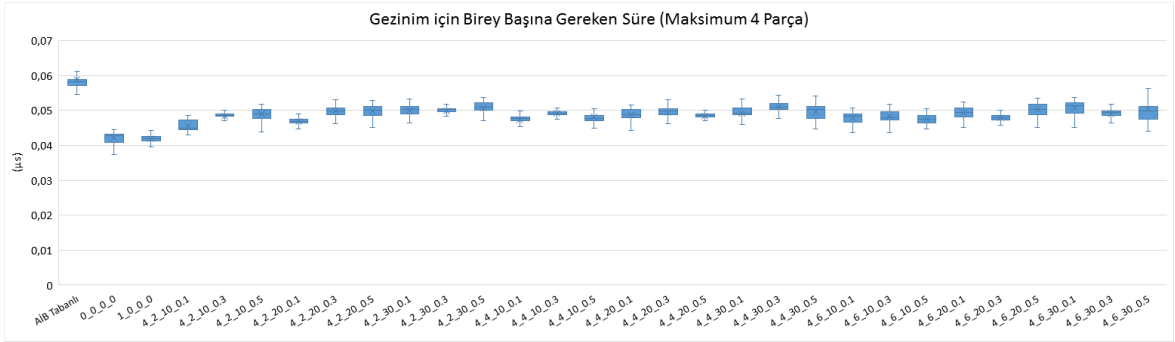
TYPÇY metodunun bireylerin benzetim alanı üzerindeki hareketlerine bağlı olarak benzetim alanı üzerine kurduğu konum-zamansal birey dizinleme yapısını güncellemesidir. Buna göre, AİB tabanlı TYPÇY metodu, her benzetim zaman adımında bireylerin pozisyon güncellemelerine ek olarak bu dizinleme yapısını güncellemek üzere fazladan işlemler gerçekleştirmektedir. Bu noktada değinilmesi gereken bir diğer önemli konu ise AİB tabanlı TYPÇY metodu da dâhil olmak üzere genel olarak geliştirilen yönlendirmesiz gezinim yöntemlerinde, birey başına gezinim maliyetinin birey sayısına bağlı olmamasıdır. Çünkü daha önce de belirtildiği üzere benzetimde var olan tüm bireyler, çevrelerindeki yoğunluk artışından bağımsız olarak kendilerine tahsis edilen çarpışmasız yollar üzerinde sadece pozisyonlarını güncellemek suretiyle ilerlemektedir. Diğer taraftan, GİB tabanlı TYPÇY yöntemi, gezinim işlemlerini gerçekleştirmek üzere ÇYPÇY yönteminin sadece TYPÇY'lere sahip bireyler oluşturmasına izin verilen benzetiminden %1'in altında daha fazla süreye ihtiyaç duymaktadır. Her iki yöntemle tekrarlanmış benzetimlerden elde edilen sonuçlar üzerinde %95 güven seviyesinde bağımsız gruplar arası t testi gerçekleştirilmiştir. Ancak bu istatistiksel analiz sonucunda gezinim maliyetleri arasındaki bu farkların istatistiksel açıdan anlamlı olmadığı tespit edilmiştir ($t(2.024) = 1.094, p = 0.387$).



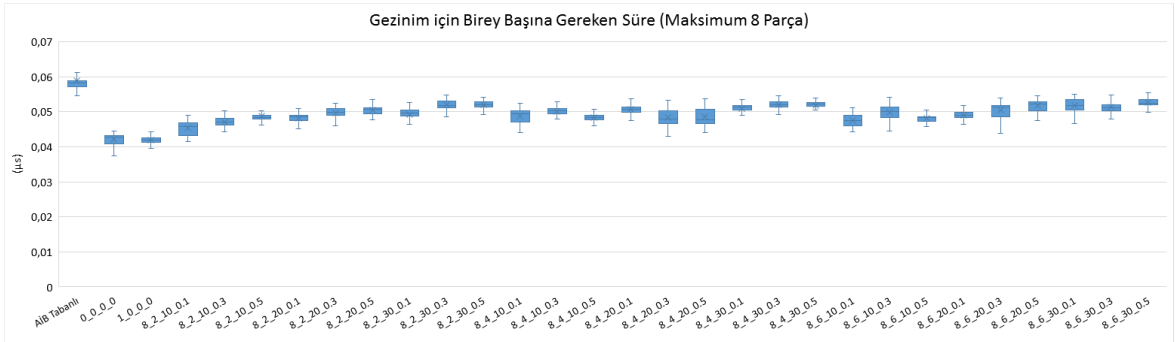
Şekil 6.52 RVO yöntemiyle gerçekleştirilen benzetim için birey başına ortalama gezinim süresi ($r = 0.5 m$)



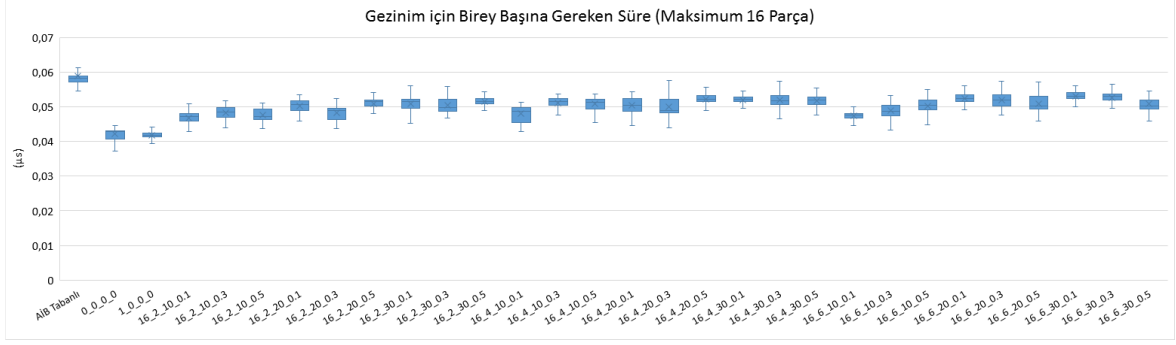
Şekil 6.53 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin farklı parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)



Şekil 6.54 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)



Şekil 6.55 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)



Şekil 6.56 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.5 m$)

ÇYPÇY yönteminin ÇYPÇY'lere sahip bireylerin gezinimi için ihtiyaç duyduğu ortalama süre, kullanılan parametre değerlerinin kombinasyonuna bağlı olarak $[0.045, 0.053] \mu s$ aralığında değişiklik göstermektedir. Bu sonuçlar göz önünde bulundurulduğunda, ÇYPÇY'lere sahip bireyler kullanan benzetimlerde gezinim için birey başına ihtiyaç duyulan ortalama sürelerin, GİB tabanlı TYPÇY yöntemine göre %7-26, ÇYPÇY yönteminin sadece TYPÇY'lere izin verilen benzetimine göre ise %8-27 artış gösterdiği görülmektedir. Bu durumun başlıca nedeni, ÇYPÇY yöntemiyle gerçekleştirilen ve ÇYPÇY'ler kullanan benzetimlerde çarpışmasız yörünge parçaları arasındaki geçiş anlarında yapılan hız vektörü ve benzeri birtakım parametrelerin güncellenmesi gibi ek işlemlerdir. Buna rağmen, her iki yöntemin de her bir bireyin gezinimini sağlamak için mikro saniyenin daha altında sürelere ihtiyaç duyduğu göz önünde bulundurulduğunda bu artış gayet normal karşılanabilmektedir. Diğer yandan, kullanılan parametre değerlerinin kombinasyonuna bağlı olarak bireylerin gezinim maliyetinde meydana gelen bu artışın değişiklik gösterdiğini söylemek mümkündür. Bu noktada ilk akla gelen, artan yörünge parçası sayısına ve yörünge parçaları arasındaki geçişte ortaya çıkan ek maliyete bağlı olarak gezinim maliyetinin artmasıdır. Ancak elde edilen sonuçlara göre, doğrudan bu yönde bir eğilimin olduğunu söylemek mümkün değildir. Yine de birey başına gezinim maliyetlerindeki bu değişikliklerin, her bir zaman adımında benzetimler içerisinde ÇYPÇY'ler kullanan birey sayısındaki farklılıkların yanı sıra bu bireylerin sahip olduğu yol parçası sayısındaki farklılıklardan ileri gelebileceği öngörülmektedir.

ÇYPÇY yönteminin ÇYPÇY'ler kullanan benzetimlerinden elde edilen veriler üzerinde birey oluşturma için harcanan ortalama süre bağımlı değişken olarak kullanılırken

max_num_of_segments, *margin_coef*, *delta_angle* ve *delta_speed* parametrelerinin bağımsız değişkenler olarak ele alındığı dört yönlü bir ANOVA gerçekleştirilmiştir. Çoklu karşılaştırma analizleri için Tukey HSD testi kullanılmış ve tüm bu istatistiksel analizlerde güven seviyesi %95 olarak ayarlanmıştır. Analiz sonuçlarına göre *max_num_of_segments* ($F(2, 72) = 13.574, p \approx 0$), *margin_coef* ($F(2, 72) = 5.973, p < 0.004$), *delta_angle* ($F(2, 72) = 49.206, p \approx 0$) ve *delta_speed* ($F(2, 72) = 5.771, p < 0.005$) parametrelerinin gezinim için birey başına harcanan ortalama süre üzerinde ana etki sahibi olduğu belirlenmiştir (Çizelge 6.11). Bu bağımsız değişkenlerin etki büyüklükleri sırasıyla; *delta_angle* için $\eta^2 = 0.445$, *max_num_of_segments* için $\eta^2 = 0.123$, *margin_coef* için $\eta^2 = 0.054$ ve *delta_speed* için $\eta^2 = 0.052$ şeklindedir (Çizelge 6.11). Çoklu karşılaştırma testlerinin sonuçlarına göre maksimum yörünge parçası sayısının 8 ($p < 0.014$) veya 16 ($p \approx 0$) olarak seçilmesi, 4 olarak seçilmesine kıyasla gezinim için birey başına harcanan ortalama sürede artışa neden olmaktadır. Benzer biçimde, yörünge parçalarının bitim noktalarının geri çekilme payı olarak $6 * r$ veya $4 * r$ tercih edilmesi, $2 * r$ seçilen benzetimlere kıyasla birey başına ortalama gezinim sürelerinde artışa neden olmaktadır ($6 * r$ için $p < 0.006$ ve $4 * r$ için $p < 0.020$). Ayrıca gezinim için birey başına harcanan ortalama süre, yörünge parçalarının birleşim noktalarında yürüme doğrultusu için izin verilen maksimum açı değişikliği ile doğru orantılıdır (tüm durumlar için $p \approx 0$). Son olarak, yörünge parçaları arasında en fazla 0.3 m/s ($p < 0.037$) veya 0.5 m/s ($p < 0.005$) yürüme hızı değişikliklerine izin vermek, 0.1 m/s yürüme hızı değişikliklerine izin verilen benzetimlere göre birey başına gezinim maliyetinde artışa neden olmaktadır.

Çizelge 6.11 ÇYPÇY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey başına gezinim süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.5 \text{ m}$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Birey başına gezinim süresi	delta_angle	2	49.206	0.000	0.445
	max_num_of_segments	2	13.574	0.000	0.123
	margin_coef	2	5.973	0.004	0.054
	delta_speed	2	5.771	0.005	0.052
	Hata	72			

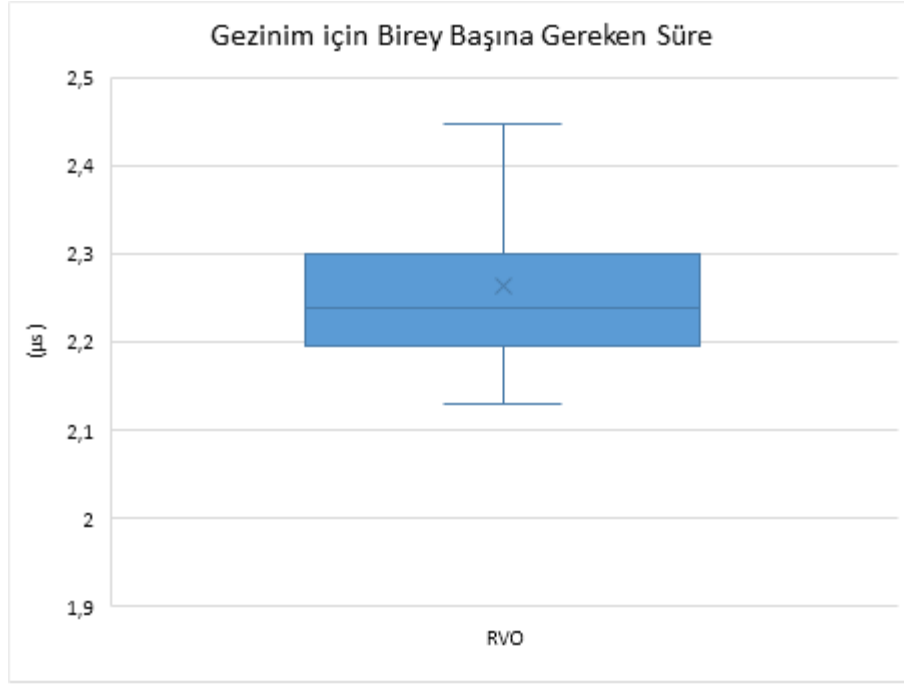
6.3.2.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar

RVO yöntemi ile gerçekleştirilen benzetimde gezinim için birey başına harcanan ortalama sürenin kutu grafiği Şekil 6.57’de sunulmuştur. RVO yöntemi, her bir bireyin gezinimini

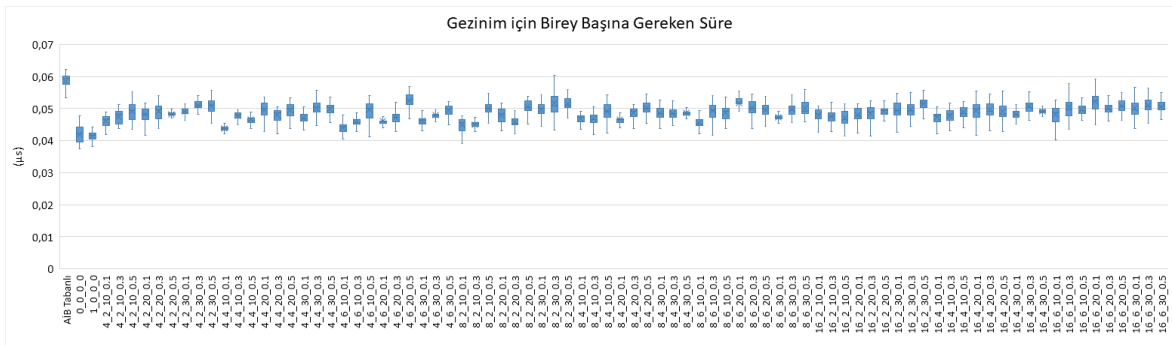
sağlamak adına yakın çevrelerini kontrol ettiği ve burada yer alan tüm diğer bireyleri göz önünde bulundurarak yeni bir çarpışmasız hız vektörü bulmaya çalıştığı için 2.263 μs değeri ile gezinim için birey başına harcanan süre konusunda en yüksek maliyete sahiptir. Diğer yöntemler ile gerçekleştirilen benzetimlerde gezinim için birey başına harcanan ortalama süreleri gösteren kutu grafikleri ise Şekil 6.58’de verilmiştir. Buna ek olarak, sonuçların yorumlanmasını kolaylaştırmak adına maksimum 4, 8 ve 16 parçalı yollar kullanan benzetim gruplarına göre bölümlenmiş grafikler, sırasıyla Şekil 6.59, Şekil 6.60 ve Şekil 6.61’de verilmiştir. Bu sonuçlar göz önünde bulundurulduğunda, AİB tabanlı TYPÇY metodunun gezinim için birey başına ortalama 0.059 μs süre gerektirdiği tespit edilmiştir. Bu değer, RVO yönteminden sonraki en yüksek birey başına gezinim süresine karşılık gelmekle birlikte RVO yöntemi, birey başına gezinim işlemleri için AİB tabanlı TYPÇY metoduna göre yaklaşık 38 kat daha fazla zamana ihtiyaç duymaktadır. GİB tabanlı TYPÇY yönteminin gezinim için birey başına harcadığı ortalama süre 0.042 μs civarındadır. Birey başına en yüksek gezinim maliyetine sahip RVO yönteminin birey başına ortalama gezinim süresinin, bu değer yaklaşık 54 katı olmasının yanı sıra AİB tabanlı TYPÇY metodu, gezinim için birey başına bu değerden yaklaşık %40 daha fazla zaman gerektirmektedir. Bununla birlikte ÇYPÇY yönteminin sadece TYPÇY’lere sahip bireylerin gezinimi için birey başına ihtiyaç duyduğu ortalama süre yaklaşık 0.042 μs olarak ölçülmüştür. Yine RVO yönteminin birey başına ortalama gezinim süresi, bu değer yaklaşık 55 katı olmakla beraber AİB tabanlı TYPÇY metodu, gezinim için birey başına bu değerden yaklaşık %41 daha fazla zamana ihtiyaç duymaktadır.

Geliştirilen tüm gezinim yöntemleri “yönlendirmesiz” olduğu için RVO yönteminden daha düşük gezinim maliyetlerine sahiptir. Buna rağmen, AİB tabanlı TYPÇY metodu, her bir zaman adımında bireylerin benzetim alanı üzerindeki hareketlerine bağlı olarak benzetim alanı üzerine kurduğu konum-zamansal birey dizinleme yapısını güncellediği için GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinden daha yüksek gezinim maliyetine sahiptir. Yine de AİB tabanlı TYPÇY metodu da dâhil olmak üzere geliştirilen yönlendirmesiz gezinim yöntemlerinin genelinde, birey başına gezinim maliyeti birey sayısına bağlı değildir. Çünkü tüm bireyler, çevrelerindeki yoğunluk artışından bağımsız olarak kendilerine tahsis edilen çarpışmasız yollar üzerinde sadece pozisyonlarını güncellemek suretiyle ilerlemektedir. Diğer taraftan, GİB tabanlı TYPÇY yöntemi, gezinim için ÇYPÇY yönteminin sadece TYPÇY’lere sahip bireyler oluşturmasına izin verilen benzetiminden %1’in altında daha fazla süre gereksinimine sahiptir. Her iki yöntemle tekrarlanmış benzetimlerden elde edilen

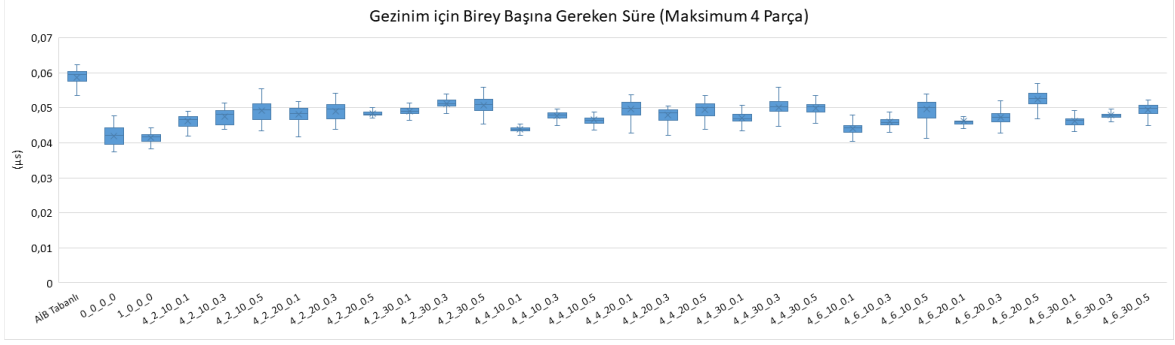
sonular zerinde %95 gven seviyesinde bağımsız gruplar arası t testi gerekleřtirilmiřtir. Ancak bu istatistiksel analiz sonucunda gezinim maliyetleri arasındaki bu farkların istatistiksel aıdan anlamlı olmadığı grlmřtr ($t(8) = 0.127, p = 0.902$).



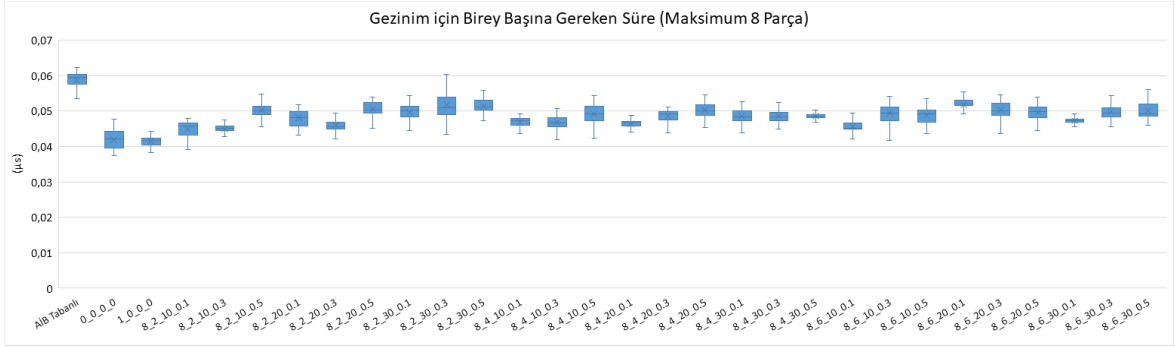
řekil 6.57 RVO yntemiyle gerekleřtirilen benzetim iin birey bařına ortalama gezinim sresi ($r = 0.3 m$)



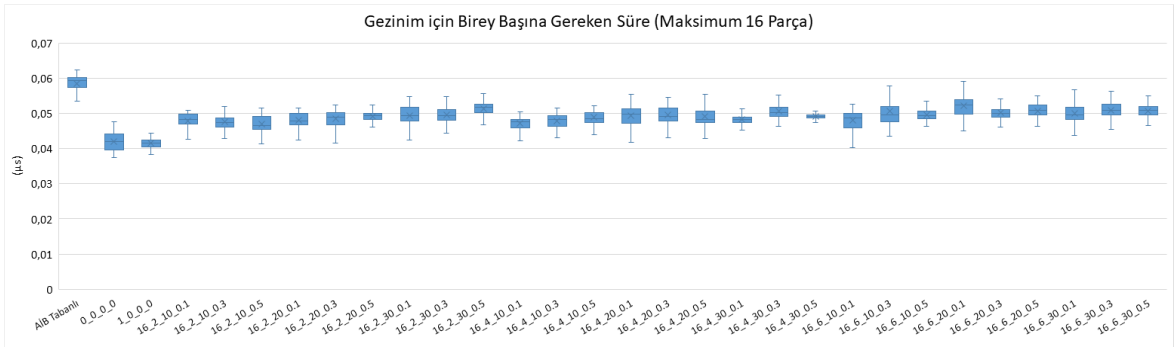
řekil 6.58 AİB ve GİB tabanlı TYPY metotlarının yanı sıra YPY yntemiyle gerekleřtirilen benzetimlerin farklı parametre kombinasyonları iin birey bařına ortalama gezinim sreleri ($r = 0.3 m$)



Şekil 6.59 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 4 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)



Şekil 6.60 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 8 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)



Şekil 6.61 AİB ve GİB tabanlı TYPÇY metotlarının yanı sıra ÇYPCY yöntemiyle gerçekleştirilen benzetimlerin maksimum 16 parçalı yollara sahip parametre kombinasyonları için birey başına ortalama gezinim süreleri ($r = 0.3 m$)

ÇYPCY yönteminde ÇYPCY'lere sahip bireylerin gezinimi için gereken ortalama süre, kullanılan parametre değerlerinin kombinasyonuna bağlı olarak [0.044, 0.052] μs aralığında yer almaktadır. Bu sonuçlar doğrultusunda, ÇYPCY'lere sahip bireyler kullanan benzetimlerde gezinim için birey başına ihtiyaç duyulan ortalama sürelerin, GİB tabanlı TYPÇY yöntemine göre %4-25, ÇYPCY yönteminin sadece TYPÇY'lere izin verilen benzetimine göre ise %5-26 artış gösterdiği tespit edilmiştir. Önceki bölümde de değinildiği üzere bu durumun temel nedeni, ÇYPCY yöntemiyle ÇYPCY'ler kullanacak şekilde gerçekleştirilen benzetimlerde çarpışmasız yörünge parçaları arasındaki geçiş anlarında yapılan hız vektörü ve benzeri birtakım parametrelerin güncellenmesi gibi ek işlemlerdir. Yine de her iki yöntemin de her bir bireyin gezinimini sağlamak için mikro saniyenin daha altında süreler gerektirdiğini hatırlatmak ve bu artışın gayet normal olduğunu vurgulamak gerekmektedir. Ayrıca kullanılan parametre değerlerinin farklı kombinasyonlarına bağlı olarak bireylerin gezinim maliyetinde meydana gelen bu artışın değişiklik gösterdiği görülmektedir. Bu değişikliğin esas nedeninin artan yörünge parçası sayısı ve yörünge parçaları arasındaki geçişte ortaya çıkan ek maliyetten daha ziyade, her bir zaman adımında benzetimler içerisinde ÇYPCY'ler kullanan birey sayısındaki farklılıkların yanı sıra bu bireylerin sahip olduğu yol parçası sayısındaki farklılıklar olduğu düşünülmektedir.

ÇYPCY yönteminin ÇYPCY'ler kullanan benzetimlerinden elde edilen veriler üzerinde birey oluşturma için harcanan ortalama süre bağımlı değişken olarak kullanılırken *max_num_of_segments*, *margin_coef*, *delta_angle* ve *delta_speed* parametrelerinin bağımsız değişkenler olarak ele alındığı dört yönlü bir ANOVA gerçekleştirilmiştir. Çoklu karşılaştırma analizleri için Tukey HSD testi kullanılmış ve tüm bu istatistiksel analizlerde güven seviyesi %95 olarak ayarlanmıştır. Analiz sonuçları incelendiğinde *max_num_of_segments* ($F(2, 72) = 5.097, p < 0.009$), *delta_angle* ($F(2, 72) = 16.932, p \approx 0$) ve *delta_speed* ($F(2, 72) = 12.274, p \approx 0$) parametrelerinin gezinim için birey başına harcanan ortalama süre üzerinde ana etki sahibi olduğu tespit edilmiştir (Çizelge 6.11). Bu bağımsız değişkenlerin etki büyüklükleri sırasıyla; *delta_angle* için $\eta^2 = 0.235$, *delta_speed* için $\eta^2 = 0.171$ ve *max_num_of_segments* için $\eta^2 = 0.071$ biçimindedir (Çizelge 6.11). Çoklu karşılaştırma testlerinin sonuçları göz önünde bulundurulduğunda, maksimum yörünge parçası sayısının 16 olarak seçilmesi, 4 olarak seçilmesine kıyasla gezinim için birey başına harcanan ortalama sürede artışa neden olmaktadır ($p < 0.006$). Ayrıca yörünge parçalarının birleşim noktalarında yürüme doğrultusunda en fazla 20° veya 30° açı değişikliklerine izin vermek, en fazla 10° doğrultu değişikliklerine izin verilen benzetimlere göre birey başına

gezinim maliyetinde artışa neden olmaktadır (tüm durumlar için $p \approx 0$). Son olarak, yörünge parçaları arasında en fazla 0.3 m/s ($p < 0.033$) veya 0.5 m/s ($p \approx 0$) yürüme hızı değişikliklerine izin vermek, 0.1 m/s yürüme hızı değişikliklerine izin verilen benzetimlere göre birey başına gezinim maliyetinde artışa neden olmaktadır.

Çizelge 6.12 ÇYPCY yönteminin birey oluşturma sürecinde kullanılan parametrelerin birey başına gezinim süresi üzerindeki etkilerinin istatistiksel analiz çıktıları ($r = 0.3 \text{ m}$)

Bağımlı Değişken	Bağımsız Değişken	df	F	p	η^2
Birey başına gezinim süresi	delta_angle	2	16.932	0.000	0.235
	delta_speed	2	12.274	0.000	0.171
	max_num_of_segments	2	5.097	0.009	0.071
	margin_coef	2	1.650	0.199	0.023
	Hata	72			

ÇYPCY yönteminin ÇYPCY'ler kullanan benzetimlerinde yarıçapı 0.5 m olan bireyler yerine yarıçapı 0.3 m olan bireyler kullanılması, birey başına gezinim maliyetini ortalama olarak %2 azaltmaktadır. Birey başına gezinim maliyetindeki bu değişiklik, %95 güven seviyesinde gerçekleştirilen bağımsız gruplar arası t testi tarafından istatistiksel açıdan anlamlı ($t(160) = 3.678$, $p \approx 0$) olarak değerlendirilmiş olmakla birlikte bu durumun ana nedeninin benzetimler içerisinde ÇYPCY'ler kullanan birey sayısındaki farklılıkların yanı sıra bu bireylerin sahip olduğu yol parçası sayısındaki farklılıklar olduğu düşünülmektedir. Bunun dışında, RVO ($t(5.374) = -0.248$, $p = 0.813$), AİB tabanlı TYPÇY ($t(6) = -0.551$, $p = 0.601$), GİB tabanlı TYPÇY ($t(6) = 0.910$, $p = 0.398$) ve ÇYPCY yöntemiyle yalnızca TYPÇY'ler ($t(6) = -0.198$, $p = 0.850$) kullanılarak gerçekleştirilen benzetimlerin her birinde, yarıçapı 0.3 m olan bireyler kullanmanın, yarıçapı 0.5 m olan bireyler kullanmaya kıyasla birey başına gezinim maliyetinde neden olduğu farklılıklar %1'in altındadır ve %95 güven seviyesinde gerçekleştirilen bağımsız gruplar arası t testlerine göre istatistiksel açıdan anlamlı değildir.

6.3.3. Bellek Maliyetlerinin Karşılaştırılması

Maliyet analizinde değinilmesi gereken bir diğer husus ise karşılaştırılan gezinim yöntemlerinin bellek ihtiyaçlarının değerlendirilmesidir. İlk olarak, RVO yönteminde her bir bireyin sadece anlık pozisyonu, hedef pozisyonu, anlık hız vektörü ve tercih ettiği hız vektörü değişkenleri için bellek alanına ihtiyacı vardır. Ayrıca bireylerin gezinim işlemleri sırasında güncel çarpışmasız hız vektörünü belirlerken göz önünde bulunduracakları yakın

çevrelerindeki diğer bireylerin daha hızlı bir biçimde tespit edilmesi adına bireyleri barındıran bir “k-d ağacı” (*k-d tree*) kullanılması da bellek tüketimi arasında sayılabilecek bir diğer başlıktır.

Diğer taraftan, AİB tabanlı TYPÇY metodu, RVO yöntemine benzer biçimde her bir birey için yalnızca bir başlangıç pozisyonu, bir bitiş pozisyonu ve bir yürüme hızı değişkeni barındırmaktadır. Ancak bu gezinim yöntemi, rastgele parametre değerleri kullanarak oluşturduğu doğrusal ve sabit hızlı yolların çarpışmasız olup olmadığını denetlemek adına bu yolların konum-zamansal uzayda var olan diğer bireylerin yolları ile kesişip kesişmediğini kontrol etmek zorundadır. Bunu yaparken de yeni oluşturulacak yolu, var olan tüm diğer yollar ile çarpışma testlerine tabi tutmak yerine sadece çarpışma ihtimali olan bireylerin yolları ile kontrol etmeyi sağlayacak konum-zamansal bir dizinleme veri yapısı geliştirilmiştir. Bu amaçla, benzetim alanı eşit büyüklükte hücrelerden oluşan bir ızgara gibi düşünülerek benzetimde yer alan bütün bireylerin başlangıç noktalarından bitiş noktalarına kadar yapacakları seyahatlerde, benzetim alanı üzerinde kapladıkları dairesel alanın temas edeceği tüm hücrelere, benzetimin başlangıcına göreli ilk temas anı ve son temas anı kaydedilmiştir. Yeni birey için oluşturulan aday yol için de 2 boyutlu benzetim alanı üzerinde temas edeceği hücreler ve bu hücreler için benzetimin başına göreli ilk temas ve son temas anları belirlendikten sonra sadece aynı hücreye aynı zaman aralığı içerisinde temas edecek bireylerin yolları ile çarpışma kontrolü yapılması sağlanmıştır. Maliyetli bir işlem olan çarpışma kontrolünün yapılacağı bireylerin sayısını önemli ölçüde azaltan bu dizinleme veri yapısı, her bir bireyin kesikli konum-zamansal bilgisini saklamayı gerektirdiği için bellek maliyetini de beraberinde getirmektedir.

Buna karşın, GİB tabanlı TYPÇY yönteminde, konum-zamansal çarpışma kontrolleri için herhangi bir dizinleme mekanizmasına gerek yoktur. Çarpışma kontrolleri, GİB'nin özel görselleştirme yetenekleri sayesinde, konum-zamansal sahnenin iz düşüm resminin yakalanması esnasında tüm bireyler için tek bir seferde hızlıca gerçekleştirildiğinden bu yöntemin ana bellek üzerindeki alan gereksinimi oldukça düşüktür. Her ne kadar GİB tabanlı TYPÇY yöntemi, bu iz düşüm resminin yakalanması ve işlenmesi için GİB üzerinde harici bellek alanları gerektiriyor olsa da gerek ihtiyaç duyulan bu bellek alanlarının boyutlarının k parametresi ile orantılı olacak biçimde ayarlanabiliyor olması gerekse GİB'nin bellek kapasitesinin çoğunlukla dar boğaz oluşturmaması, bu durumu önemli bir sorun teşkil etmekten alıkoymaktadır.

ÇYPCY yönteminde, GİB tabanlı TYPÇY yönteminin GİB üzerinde konum-zamansal sahnenin iz düşün resminin yakalanması ve işlenmesi için ihtiyaç duyduğu harici belleğe ek olarak derinlik testlerinde derinlik bilgilerinin yazılıp okunduğu derinlik belleklerine ihtiyaç vardır. Yine de derinlik bellekleri, bahsi geçen diğer işlemler için kullanılan üç bileşenli renk belleklerinin aksine yalnızca tek bileşenden oluştuğu için her bir derinlik belleği renk belleklerinin üçte biri kadar alana ihtiyaç duymaktadır. Bunun haricinde, ÇYPCY yönteminin benzetimi yapılan birey başına ihtiyaç duyduğu bellek gereksinimi GİB tabanlı TYPÇY yöntemi ile kıyaslandığında, kullanılan yörünge parçası sayısının artmasına paralel olarak artmaktadır. Buna karşın, her bir yörünge parçası bir başlangıç pozisyonu, bir bitiş pozisyonu ve bir yürüme hızı değerinden ibaret olduğu için bellek maliyetindeki bu artış çok da kayda değer değildir. Diğer taraftan, GİB tabanlı TYPÇY yönteminin her bir birey oluşturma sürecinde, bitişik beyaz doku hücreleri GPGPU geçişleriyle gruplanıyor olmasına rağmen, oluşturulan tüm grupların bilgilerinin AİB belleğine transfer ediliyor olması sebebiyle GİB tabanlı TYPÇY yönteminin sadece tek bir olası çözümün bilgilerini döndüren ÇYPCY yöntemine kıyasla daha yüksek bellek tüketimi söz konusudur.

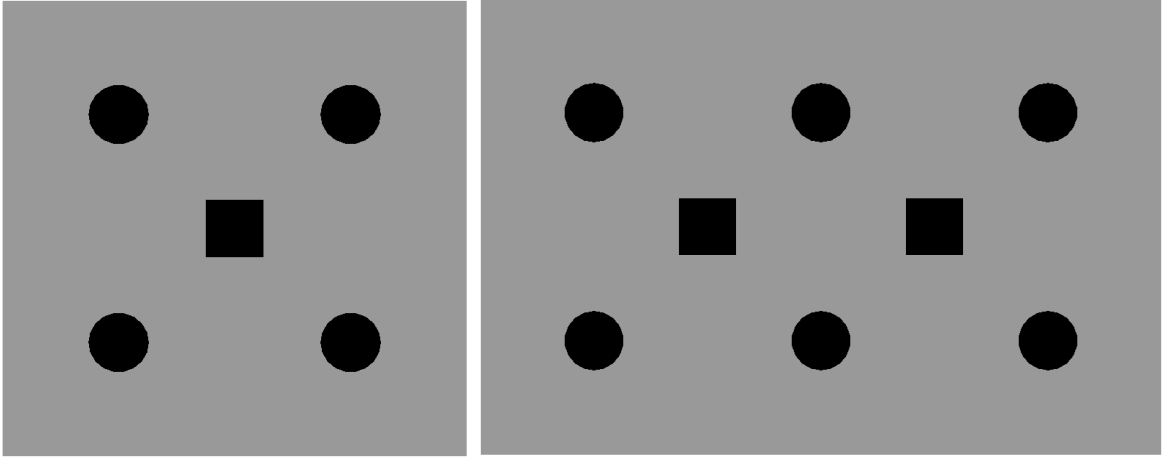
6.4. Sabit Engeller İçeren Benzetimlerdeki Birey Oluşturma Başarımının İncelenmesi

Şimdiye kadar gerçekleştirilen deneylerde kullanılan benzetim alanları üzerinde hiçbir sabit engel bulunmamaktadır. Ancak tez kapsamında geliştirilen yönlendirmesiz gezinim yöntemlerinin sabit engeller içeren benzetim alanları üzerinde de kalabalık benzetimleri gerçekleştirmesi mümkündür. AİB tabanlı TYPÇY yönteminde sabit engeller ile çarpışmayacak bireyler oluşturulması için bu engellerin 2 boyutlu benzetim alanı üzerinde temas ettiği konum-zamansal dizinleme yapısının hücrelerine bazı ek bilgiler eklenmelidir. Buna göre, sabit engellerin temas ettiği hücreleri ziyaret edecek biçimde oluşturulan aday yörüngeler, çarpışmasız olduklarını garanti etmek adına sadece daha önce anlatıldığı gibi aynı anda aynı hücrede bulunacağı için çarpışma ihtimali bulunan bireyler ile değil, bir kısmı veya tamamı bu hücreler içerisinde bulunan sabit engeller ile de kesin çarpışma testlerine tabi tutulmalıdır. Dairesel bir sınırlayıcı alana sahip olan sabit engeller ile gerçekleştirilecek kesin çarpışma testleri, bireyler arasında yapılanlara çok benzer olmakla birlikte yalnızca pozisyonlar arasındaki mesafenin karşılaştırıldığı eşik değerini iki bireyin yarıçapları toplamı değil de bireyin ve sabit engelin yarıçapları toplamı kadar olması gerekmektedir. Diğer yandan, özellikle sınırlarının merkezden uzaklığı her noktada aynı olmayan farklı bir geometrik şekildeki sınırlayıcı alana sahip olan sabit engeller için daha karmaşık hesaplamalar içeren farklı test yöntemleri kullanılması gerekmektedir. Bu noktada

kullanılabilecek yaklaşımlardan biri, sabit engelin sınırlayıcı alanının çevresini doğru parçalarına ayırarak bu doğru parçalarından her biri ile yeni oluşturulacak bireyin ziyaret edeceği ızgara hücrelerini belirlemede kullanılan, bireyin yörüngesine paralel ve diskinde teğet yardımcı iki doğru parçası arasında kesişim testleri uygulamaktır. Eğer yeni oluşturulacak bireyin doğru parçaları ile sabit engelin doğru parçalarının hiç biri kesişmiyor ise bireyin bu sabit engel ile çarpışmayacağı söylenebilir. Tam tersine bu kesişim testlerinden birinde bir kesişim tespit edilirse, birey bu yörünge ile sabit engele çarpacağı için yeni bir yörünge ile yeni bir birey oluşturma denemesi gerçekleştirilmelidir.

Diğer taraftan, geliştirilen GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin çarpışmasız yollara sahip birey oluşturma sürecinde sabit engelleri de göz önünde bulundurmamak nispeten daha kolaydır. Sabit engeller herhangi bir hıza sahip olmadığı ve hareket etmediği gibi benzetim alanı üzerinde kapladıkları alanlar da yer değiştirmeyecektir. Dolayısıyla konum-zamansal uzayda bu alandan dolayı kapladıkları dik silindir veya prizma biçimindeki hacimler, tabanı benzetim alanı düzlemi üzerinde yer alan ve zaman ekseninde sonsuza kadar uzanan kapalı 3 boyutlu cisimlere (KZYE) karşılık gelmektedir. Benzetimde kullanılacak sabit engellerin benzetim alanı üzerindeki izdüşümünü içine alacak biçimde uygun bir dairesel veya çokgensel alan belirlendikten sonra, tabanı bu alan ile eş biçimli olan uygun 3 boyutlu dik silindir veya prizma modellerinin temin edilmesi gerekmektedir. Bu aşamadan sonra her bir birey oluşturma denemesinde, benzetimde var olan bireylerin konum-zamansal görselleştirilmesinin yanı sıra konum-zaman uzayında bu sabit engellere karşılık gelen silindir veya prizma biçimindeki hacimlerin de görselleştirilmesi gerçekleştirilerek çarpışmasız yol belirleme sürecinde bu engellerin de gayet kolay bir biçimde göz önünde bulundurulması sağlanabilir.

TYPÇY'ler ve ÇYPÇY'ler kullanan yönlendirmesiz gezinim yöntemlerinin sabit engeller içeren benzetim alanları üzerinde birey oluşturma başarımını incelemek üzere birden fazla sabit engel içeren iki farklı benzetim alanı oluşturulmuştur. Bunlardan ilki, bir kenarının uzunluğu 128 *m* olan kare biçiminde bir benzetim alanı (Şekil 6.62 (Sol)), ikincisi ise kısa kenarının uzunluğu 128 *m* ve uzun kenarının uzunluğu 192 *m* olan dikdörtgen biçiminde bir benzetim alanıdır (Şekil 6.62 (Sağ)). Her iki benzetim alanı üzerinde de daire ve kare biçimde, farklı sayılarda sabit engeller yer almaktadır (Şekil 6.62).

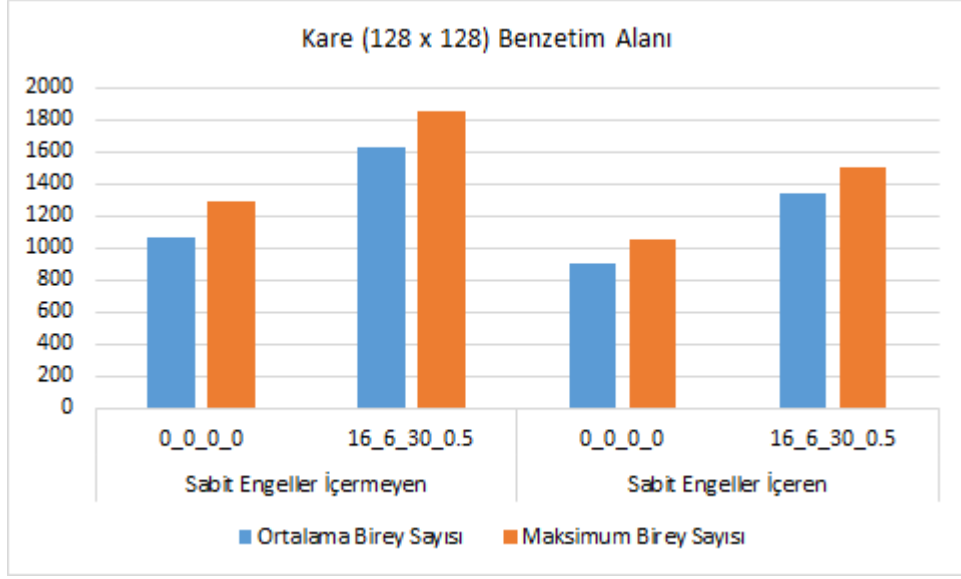


Şekil 6.62 (Sol) Bir kenarı $128\ m$ olan kare biçiminde bir benzetim alanı, (Sağ) kısa kenarı $128\ m$ ve uzun kenarı $192\ m$ olan dikdörtgen biçiminde bir benzetim alanı ve bu alanlar (gri) üzerinde yer alan daire ve kare biçimindeki sabit engeller (siyah)

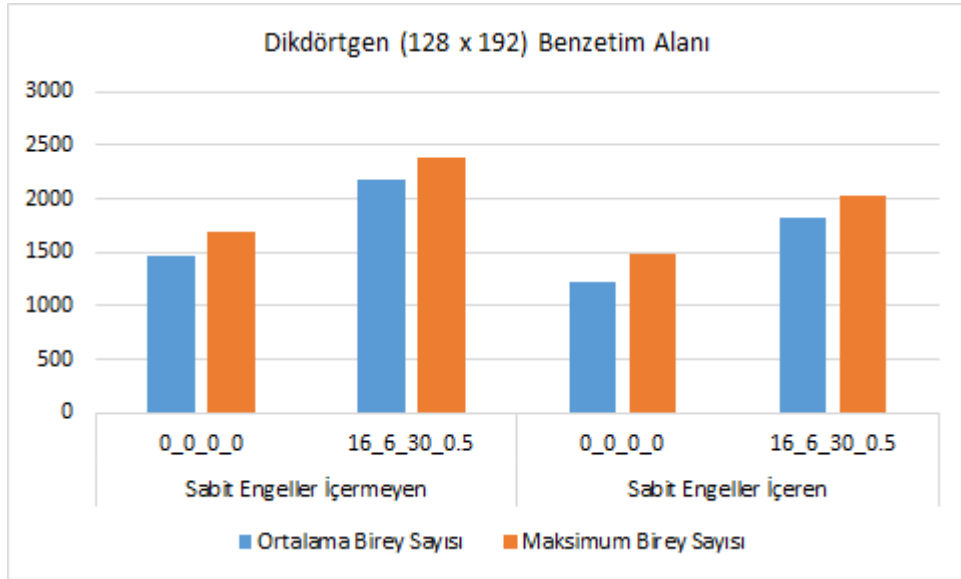
Daha önceki bölümlerde verilen birey oluşturma başarımları deneylerinde, sabit engel içermeyen benzetim alanları üzerinde en yüksek birey sayılarına ulaşmaları nedeniyle, tanımlanan bu benzetim alanları üzerinde TYPÇY yaklaşımını temsilen GİB tabanlı TYPÇY yöntemi ($0_0_0_0$) ve ÇYPÇY yaklaşımını temsilen ÇYPÇY yönteminin en geniş çözüm uzayı sunan parametre kombinasyonu ($16_6_30_0.5$) ile 5 dakika uzunluğunda benzetimler çalıştırılmıştır. Her iki gezinim yöntemi ile yapılan benzetimlerde de birey oluşturma işlemleri için her bir benzetim zaman adımında sabit $25\ ms$ süre ayrılmış ve bu süre zarfında başarılı veya başarısız olduğuna bakmaksızın benzetimlerin art arda mümkün olduğunca çok birey oluşturma denemesi yapması sağlanmıştır. Bu benzetimler boş bir benzetim alanı ile başlayıp bu alanı mümkün olduğunca fazla sayıda bireyle doldurmak için çalışmaktadır. Benzetimlerin her bir saniyesinde benzetimde yer alan bireylerin sayısı ölçülerek benzetim sonunda ortalama ve maksimum birey sayıları hesaplanmıştır. Ayrıca karşılaştırma sağlamak adına her iki yöntem ile tanımlanan bu benzetim alanlarının eşit büyüklükte olan, ancak hiçbir sabit engel içermeyen benzerleri üzerinde de aynı özelliklere sahip benzetimler çalıştırılmış ve bu benzetimlerin her bir saniyesinde de benzetimde yer alan bireylerin sayısı ölçülerek benzetim sonunda ortalama ve maksimum birey sayıları hesaplanmıştır. Tüm bu deneyler, birey yarıçapının $0.5\ m$ ve $0.3\ m$ olmak üzere iki farklı değeri için iki farklı grup olarak tekrarlanmıştır.

6.4.1. Birey Yarıçapının 0.5 m Değeri için Sonuçlar

Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayılarını gösteren grafik Şekil 6.63'te verilmiştir. Benzer biçimde, dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayılarını gösteren grafik ise Şekil 6.64'te verilmiştir. Sonuçlar incelendiğinde GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin her ikisinin de hem kare hem de dikdörtgen benzetim alanlarının sabit engeller içeren türevleri üzerindeki benzetimlerinde elde edilen ortalama ve maksimum birey sayılarının, sabit engeller içermeyen türevleri üzerindeki benzetimlerinde elde edilen ortalama ve maksimum birey sayılarından daha düşük olduğu görülmektedir. GİB tabanlı TYPÇY yönteminin kare benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine kıyasla ortalamada %16 ve maksimumda %18 daha az birey oluşturabilmektedir. Benzer biçimde, GİB tabanlı TYPÇY yönteminin dikdörtgen benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine kıyasla ortalamada %17 ve maksimumda %12 daha az birey içermektedir. Diğer taraftan, ÇYPÇY yönteminin kare benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine kıyasla ortalamada %18 ve maksimumda %19 daha az birey oluşturabilmektedir. Aynı şekilde, ÇYPÇY yönteminin dikdörtgen benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine kıyasla ortalamada %17 ve maksimumda %15 daha az birey içermektedir. Benzetim alanları üzerinde yer alan sabit engellerin konum-zamansal uzayda sürekli hacim işgal ederek olası çözümlerin sayısını azaltması nedeniyle ortalama ve maksimum birey sayılarındaki bu düşüşler gayet normaldir.



Şekil 6.63 Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.5 m$)



Şekil 6.64 Dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.5 m$)

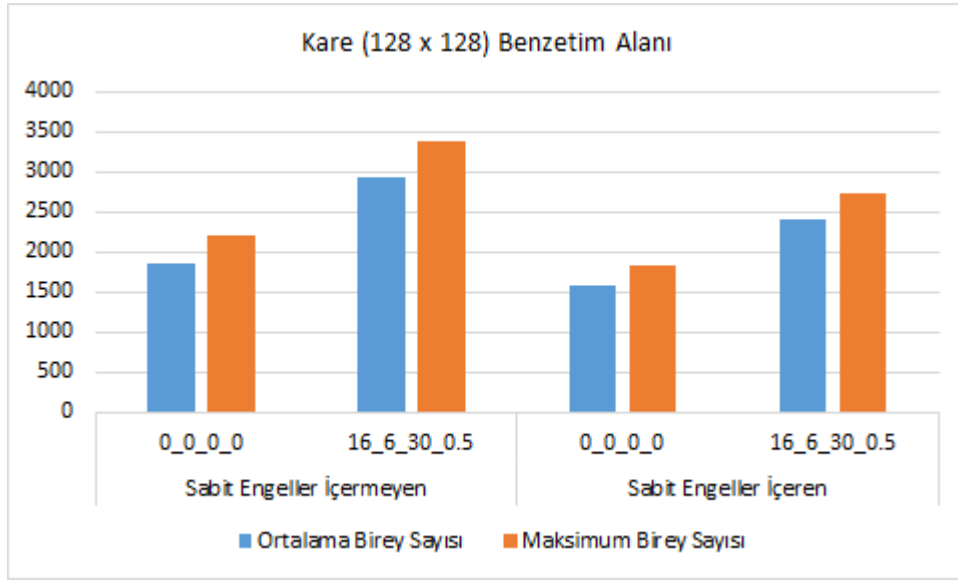
Diğer yandan, bu sonuçlara bakılarak ÇYPCY yönteminin her durumda GİB tabanlı TYPÇY yönteminden daha yüksek ortalama ve maksimum birey sayılarına ulaşabildiğini gözlemlemek de mümkündür. ÇYPCY yöntemiyle kare benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetim, GİB tabanlı TYPÇY yöntemiyle aynı

benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetimden daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %48 ve %43) sahip olmakla kalmayıp GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içermeyen türevi üzerinde gerçekleştirilen ve dolayısıyla daha yüksek birey sayılarına ulaşabilen benzetimden de daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %25 ve %17) sahiptir. Benzer şekilde, ÇYPCY yöntemiyle dikdörtgen benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetim, GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetimden daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %49 ve %38) sahip olmakla kalmayıp GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içermeyen türevi üzerinde gerçekleştirilen ve dolayısıyla daha yüksek birey sayılarına ulaşabilen benzetimden de daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %23 ve %20) sahiptir. ÇYPCY yönteminde ÇYPCY'lerin kullanımına olanak tanınmasından dolayı genişleyen olası çözüm uzayı, içerisine ilave edilen sabit engeller nedeniyle bir miktar küçülerek olası çözümlerin sayısında düşüşe yol açmaktadır. Buna rağmen, yalnızca TYPÇY'ler kullanabilen GİB tabanlı TYPÇY yönteminin, sabit engeller içerdiği için bir miktar küçülmüş olan olası çözüm uzayından daha geniş olmakla kalmayıp sabit engeller içermeyen maksimum büyüklükteki olası çözüm uzayından da daha geniştir. Bu nedenle her durumda daha fazla olası çözüme sahip olan ÇYPCY yöntemi, aynı benzetim alanlarının sabit engeller içeren türevleri üzerinde dahi daha yüksek birey sayılarına ulaşabilmektedir.

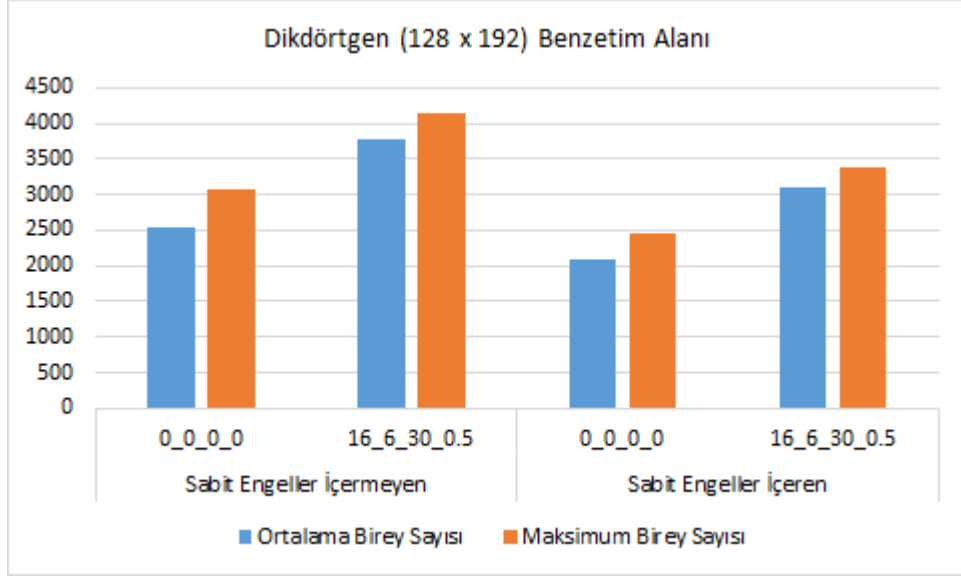
6.4.2. Birey Yarıçapının 0.3 m Değeri için Sonuçlar

Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayılarını gösteren grafik Şekil 6.65'te verilirken dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayılarını gösteren grafik ise Şekil 6.66'da verilmiştir. Bu sonuçlar göz önünde bulundurulduğunda, GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin her birinin kare ve dikdörtgen benzetim alanlarının sabit engeller içeren türevleri üzerindeki benzetimlerinde elde edilen ortalama ve maksimum birey sayılarının, sabit engeller içermeyen türevleri üzerindeki benzetimlerinde elde edilen ortalama ve maksimum birey sayılarından daha düşük olduğu gözlemlenmektedir. GİB tabanlı TYPÇY yönteminin kare benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine göre ortalama %15 ve maksimumda %17 daha az bireye sahiptir. Aynı şekilde, GİB tabanlı TYPÇY yönteminin

dikdörtgen benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine göre ortalamada %17 ve maksimumda %19 daha az birey oluşturabilmektedir. Ayrıca ÇYPÇY yönteminin kare benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine göre ortalamada %17 ve maksimumda %19 daha az birey içermektedir. Benzer olarak, ÇYPÇY yönteminin dikdörtgen benzetim alanının sabit engeller içeren türevi üzerindeki benzetimi, aynı benzetim alanının sabit engeller içermeyen türevine göre ortalamada %18 ve maksimumda %18 daha az bireye sahiptir. Ortalama ve maksimum birey sayılarındaki bu düşüşlerin ana nedeni, benzetim alanları üzerinde yer alan sabit engellerin konum-zamansal uzayda sürekli hacim işgal ederek olası çözümlerin sayısını azaltmasıdır.



Şekil 6.65 Kare benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.3 m$)



Şekil 6.66 Dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen türevleri üzerinde gerçekleştirilen benzetimlerin ortalama ve maksimum birey sayıları ($r = 0.3 m$)

Tüm bunlara ek olarak, bu sonuçlar incelendiğinde ÇYPCY yönteminin her durumda GİB tabanlı TYPÇY yönteminden daha yüksek ortalama ve maksimum birey sayılarına ulaşabildiği söylenebilir. ÇYPCY yöntemiyle kare benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetim, GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetimden daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %54 ve %49) sahip olmanın yanı sıra GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içermeyen türevi üzerinde gerçekleştirilen ve dolayısıyla daha yüksek birey sayılarına ulaşabilen benzetimden de daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %31 ve %24) sahiptir. Aynı biçimde, ÇYPCY yöntemiyle dikdörtgen benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetim, GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içeren türevi üzerinde gerçekleştirilen benzetimden daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %48 ve %37) sahip olmakla birlikte GİB tabanlı TYPÇY yöntemiyle aynı benzetim alanının sabit engeller içermeyen türevi üzerinde gerçekleştirilen ve dolayısıyla daha yüksek birey sayılarına ulaşabilen benzetimden de daha yüksek ortalama ve maksimum birey sayılarına (sırasıyla %22 ve %10) sahiptir. ÇYPCY yönteminde ÇYPCY'lerin kullanımına olanak tanınmasından dolayı genişleyen olası çözüm uzayı, içerisine ilave edilen sabit engeller nedeniyle bir miktar küçülerek olası çözümlerin sayısında düşüşe yol açmaktadır. Ancak bu durumuyla bile sadece TYPÇY'ler kullanabilen

GİB tabanlı TYPÇY yönteminin hem sabit engeller içerdiği için bir miktar küçülmüş olan olası çözüm uzayından hem de sabit engeller içermeyen maksimum büyüklükteki olası çözüm uzayından daha geniştir. Bundan dolayı her durumda daha fazla olası çözüme sahip olan ÇYPÇY yöntemi, aynı benzetim alanlarının sabit engeller içeren türevleri üzerinde bile daha yüksek sayıda birey içerebilmektedir.

Birey oluşturma başarımını inceleyen deneylerin sonuçlarına benzer şekilde, birey yarıçapının 0.5 m yerine 0.3 m olarak kullanılması, her iki gezinim yöntemiyle de hem kare hem de dikdörtgen benzetim alanının sabit engeller içeren ve içermeyen tüm türevleri üzerinde daha fazla birey oluşturulmasını olanaklı kılmaktadır. Elde edilen sonuçlara göre, kare benzetim alanının sabit engeller içermeyen türevi üzerinde GİB tabanlı TYPÇY yöntemi ortalamada %73, maksimumda %72 daha fazla birey oluşturabilirken ÇYPÇY yöntemi ortalamada %79, maksimumda %82 daha fazla bireye sahip olabilmektedir. Aynı şekilde, kare benzetim alanının sabit engeller içeren türevi üzerinde ise GİB tabanlı TYPÇY yöntemi ortalamada %74, maksimumda %75 daha fazla bireye yer verebilirken ÇYPÇY yöntemi ortalamada %81, maksimumda %82 daha fazla bireye ulaşabilmektedir. Diğer yandan, dikdörtgen benzetim alanının sabit engeller içermeyen türevi üzerinde GİB tabanlı TYPÇY yöntemi ortalamada %72, maksimumda %81 daha fazla birey oluşturabilirken ÇYPÇY yöntemi ortalamada %72, maksimumda %73 daha fazla bireye sahip olabilmektedir. Benzer biçimde, dikdörtgen benzetim alanının sabit engeller içeren türevi üzerinde ise GİB tabanlı TYPÇY yöntemi ortalamada %71, maksimumda %67 daha fazla bireye yer verebilirken ÇYPÇY yöntemi ortalamada %70, maksimumda %66 daha fazla bireye ulaşabilmektedir.

6.5. Geliştirilen Gezinim Yöntemlerinin Yörünge Kalitesinin Değerlendirilmesi

Daha önce de belirtildiği üzere tez kapsamında geliştirilen gezinim metotları yönlendirmesiz gezinim yaklaşımını benimsemektedir. Bu yöntemlerde bireylere benzetime dâhil olmalarından hemen önce, başlangıç noktalarından bitiş noktalara kadar uzanan ve yalnızca benzetimde yer alan diğer bireyler ile değil, bir önceki bölümde örneklendiği üzere benzetim alanı üzerinde bulunan sabit engeller ile de çarpışmasız olduğu garanti edilen yollar tahsis edilmektedir. Benzetim boyunca bireyler, hiçbir şekilde çarpışma tespiti ve çarpışmadan kaçınma manevrası yapmak üzere hesaplama gerçekleştirmelerine gerek olmadan kendilerine atanan çarpışmasız yollar üzerinde sabit hızlar ile yürümektedir. Bu gibi bir hareketin ilk bakışta oldukça monoton ve gerçek dışı gözükeceği öngörülebilir. Ancak “en az çaba prensibi” doğrultusunda; genel olarak tüm canlıların [10], daha özelde yürüyen

insanların [11] ve tez çalışmasının motivasyonu ile doğrudan ilgili olarak kalabalığı oluşturan bireylerin [12] hedeflerine ulaşmak üzere en az çaba (minimum enerji) gerektireceklerini düşündükleri yolları kullanmayı seçerek buna uygun dinamikler ve davranışlar sergileyeceği ileri sürülmekle birlikte çeşitli deney ve gözlemlerle de bu iddia desteklenmiştir. Bu bilgi ve bulgular ışığında, geliştirilen yönlendirmesiz gezinim yöntemlerinde doğrusal çarpışmasız yörünge parçaları üzerinde manevra yapmaksızın sabit hızlarla yürüyen bireylerin, öngörülenin aksine inandırıcılığı olumsuz etkilemeyeceğini savunmak da mümkündür.

Nitekim gezinim esnasında manevra yapmak, beklenilenin aksine her zaman benzetimin inandırıcılığını artırmayabilir. Örneğin, alanda var olan küçük ölçekli gezinim modellerinin birçoğu, bireylerin yalnızca sınırlı bir mesafedeki yakın çevrelerini değerlendirerek gelecekteki olası çarpışmalardan kaçınmalarını sağlayacak manevralar yapmak üzere hız vektörlerini güncellediği için hatalı veya eksik gezinim kararları vermektedir. Bu kararlar neticesinde gerçekleştirilen manevralar ise çoğu zaman hedefe doğru ilerledikleri yörüngeler üzerinde kararsız bir biçimde sürekli devam eden bir sağa bir sola salınım hareketlerine neden olmaktadır. Bunun yanı sıra, kimi zaman karşı yönden kendisine doğru gelen yoğun bir birey grubu nedeniyle hedefine doğru ilerlemek üzere geçerli bir çarpışmasız hız vektörü belirleyemeyen bireylerin; istemsiz bir biçimde geriye doğru döndüğü, karşı yönden gelen bu grubun akıntısına kapılarak bu gruptaki diğer bireyler ile aynı yöne doğru sürüklendiği ve ancak belirli bir süre sonra bu grubun etkisinden kurtularak tekrar hedefine doğru yöneldiği görülmektedir. Bazen de hedeflerine doğru olan hareketleri esnasında karşı karşıya gelen birey veya birey grupları, birbirleri için verilen gezinim kararlarından kısmen veya tamamen haberdar olmadıkları için sürekli değişen kararları sebebiyle birbirlerine girmekte ve bu sıkışık grup içerisinde bir süre oldukları yerde sağa sola dönerek çıkış yolu aramak suretiyle takılı kalmaktadır. Manevra kabiliyeti daha yüksek olmasına karşın kimi zaman bireylere istikrarlı ve pürüzsüz bir hareket sağlayamayan bu gibi gezinim yöntemlerinin algılanan gerçekçiliği daha olumsuz etkileyeceği aşikârdır.

Bu durumu örneklemek adına RVO ve ÇYPCY yöntemleri ile bir kenarının uzunluğu 32 m olan kare biçiminde benzetim alanı üzerinde iki ayrı benzetim gerçekleştirilmiştir. Benzetim alanının önceki deneylerin aksine daha küçük seçilmesinin nedeni, bireylerin ve izledikleri yörüngelerin daha yakın bir bakış açısından görüntülenmek istenmesidir. ÇYPCY yöntemiyle gerçekleştirilen benzetimde en yüksek ortalama birey sayısı değerine ulaşabilen parametre kombinasyonu (16_6_30_0.5) kullanılmasına rağmen, bireylerin benzetim alanı

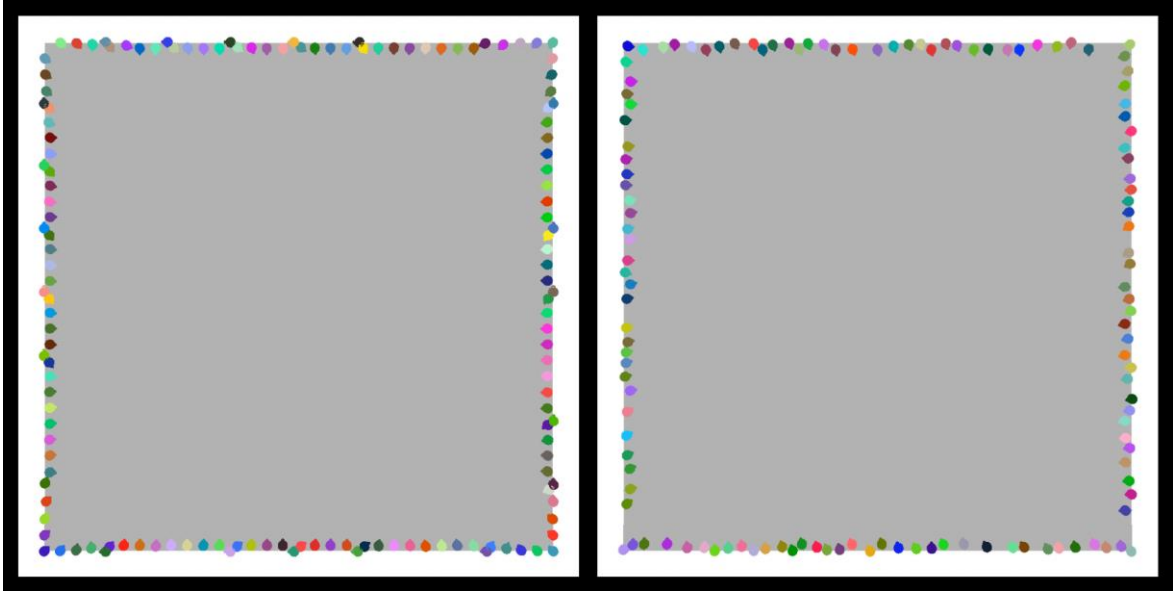
üzerindeki yörünge izlerini takip edebilmek adına her iki benzetimde de maksimum birey sayıları 300 olarak sınırlanmıştır. Bu sayede her iki benzetim boyunca ortalama birey sayılarının yaklaşık 300 civarında olması sağlanmıştır.

Bu benzetimlerin başlangıcından itibaren 10 saniyelik aralıklarla yakalanan anlık kuşbakışı görüntüleri, zaman sırasına göre Şekil 6.67, Şekil 6.68, Şekil 6.69, Şekil 6.70, Şekil 6.71, Şekil 6.72 ve Şekil 6.73'te verilmiştir. Bu görüntüler incelendiğinde, benzetimin başlangıcında RVO yönteminin, ÇYPÇY yöntemine göre daha fazla bireyi hızlı bir biçimde oluşturabildiği görülmektedir (Şekil 6.67). Benzetimlerin devamında ise RVO yöntemi bireylerin benzetim alanının merkezinde çok dar bir alanda birikerek sıkışmasına neden olmaktadır (Şekil 6.68). Bir girdap gibi dönerek zaman içerisinde çözülen bu takılmadan kurtulan bireyler, hedeflerine doğru yollarına devam edebilmektedir (Şekil 6.69 ve Şekil 6.70). ÇYPÇY yönteminin ise benzetimin başlangıcından itibaren çok daha düzgün dağılımlı bir kalabalık oluşturmayı başarabildiği görülmektedir (Şekil 6.67, Şekil 6.68, Şekil 6.69 ve Şekil 6.70).

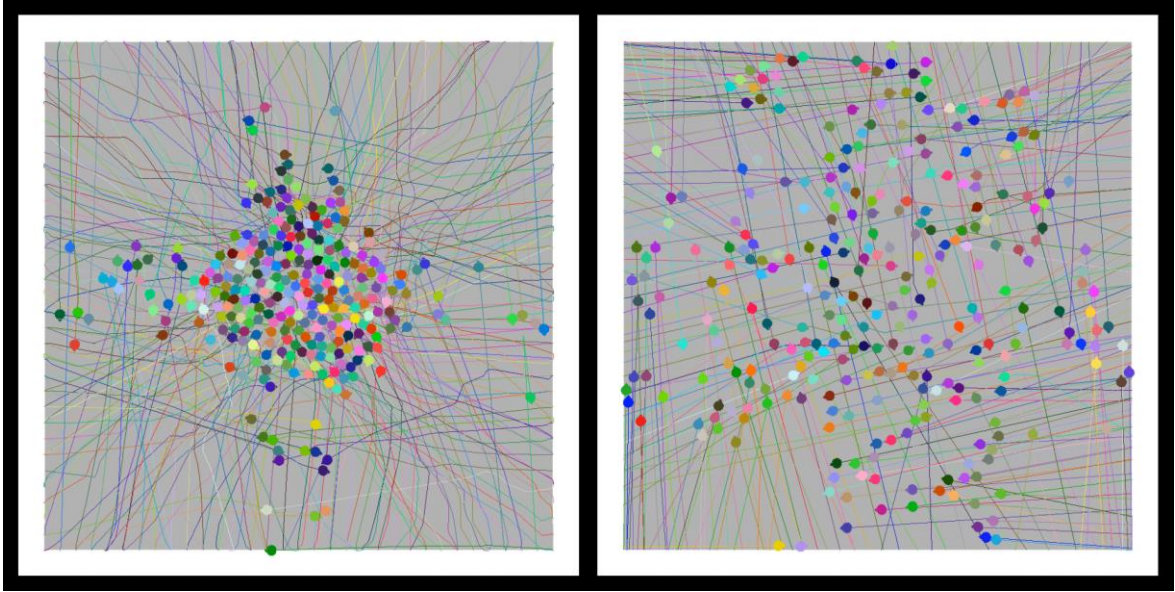
Benzetim alanının ortasında meydana gelen yoğunluk çözüldükten sonra RVO yöntemi, öncekine göre daha düzgün bir birey dağılımına ulaşabilmektedir. Ancak bu sefer de benzetim alanının kenarlarına yakın bölgelerinde hedeflerine varmak üzere olan bireyler ile bu kenarlar üzerinden benzetime yeni katılan veya bu kenarlara yakın ve paralel biçimde ilerleyen bireyler arasında takılmalar yaşandığı gözlemlenmektedir (Şekil 6.71, Şekil 6.72 ve Şekil 6.73). Verilen görsellerde yer alan bireylerin kendileriyle aynı renkte gösterilen geçmiş yörünge izlerinden de anlaşılacağı üzere RVO ile gerçekleştirilen benzetim içerisinde yer alan ve özellikle takılma yaşayan bireylerin bu durumdan kurtulmak adına yukarıda örneklenen sürekli salınımlı manevralar, ani geri dönüşler gibi tutarsız ve pürüzlü hareketler gerçekleştirdiğini de belirtmek gerekir. Diğer taraftan, ÇYPÇY yöntemi, benzetimin ilerleyen kesiminde de RVO yöntemine kıyasla daha düzgün dağılımlı bir kalabalık oluşturmaya devam etmektedir (Şekil 6.71, Şekil 6.72 ve Şekil 6.73). Bunun yanı sıra, ÇYPÇY yönteminde yer alan bireyler, yalnızca yörünge parçaları arasındaki geçiş noktalarında, aynı şekilde harekete devam etmeleri durumunda çarpışmaya neden olacak eski hız vektörlerini bu çarpışmadan kaçınmayı sağlayacak yeni hız vektörleri ile güncelleyerek bir nevi çarpışmadan kaçınma manevraları gerçekleştirmektedir. Ancak burada sözü edilen manevralar, yalnızca gerekli olduğu zamanda ve gerekli olduğu ölçüde yapıldığı için bireylerin tutarlı ve pürüzsüz hareketine katkı sağlamakla birlikte bölümün

başında yer verilen yönlendirmesiz gezinimle ilgili öngörülerin ve RVO yöntemiyle gerçekleştirilen benzetimlerin aksine çok daha tatmin edici benzetimler sunmaktadır.

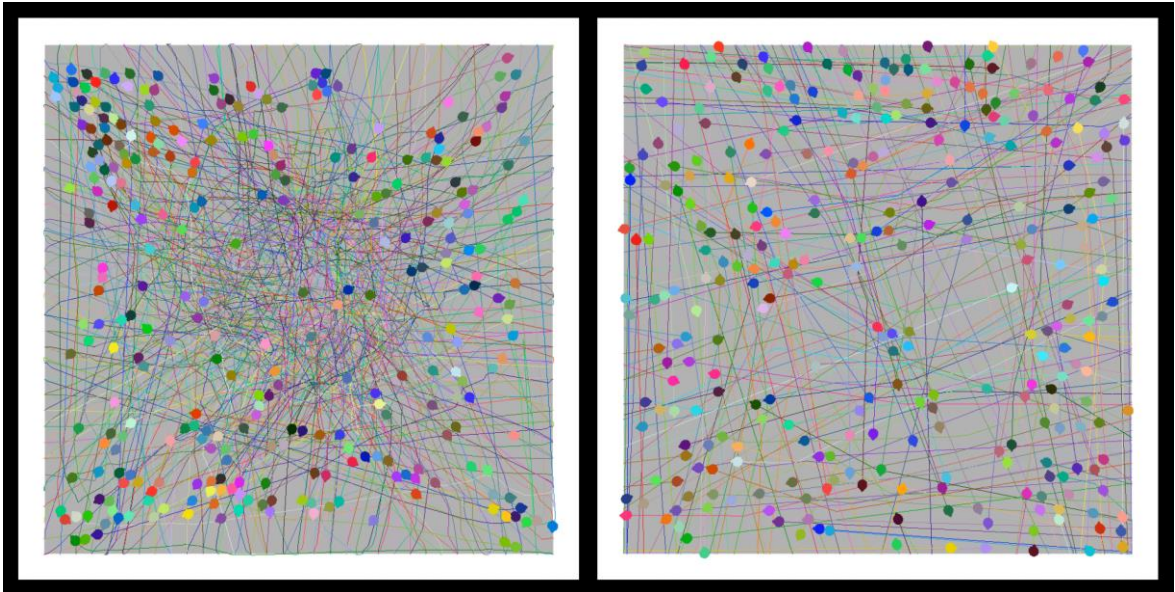
Geliştirilen yönlendirmesiz gezinim yöntemleri arasında en yüksek ortalama birey yoğunluğuna ulaşabilmesi nedeniyle bu deneylerde ÇYPCY yöntemi kullanılmıştır. Ancak burada elde edilen sonuçların TYPÇY veya ÇYPCY kullanımından bağımsız olarak geliştirilen diğer yöntemlerle veya bunların farklı parametre kombinasyonlarıyla gerçekleştirilecek benzetimlere de genellenebilir olduğunu belirtmekte fayda vardır. Çünkü bu yöntemlerin daha düşük birey sayılarına sahip benzetimleri ile yapılacak farklı karşılaştırmalarda, RVO yöntemiyle yapılacak benzetimlerde birey yoğunluğundaki azalmaya bağlı olarak istenmeyen davranışlar ile karşılaşma ihtimali de azalsa bile geliştirilen tüm yönlendirmesiz gezinim yöntemleri birey sayısından bağımsız olarak bireylerin tutarlı ve pürüzsüz hareket kabiliyetini koruyacaktır.



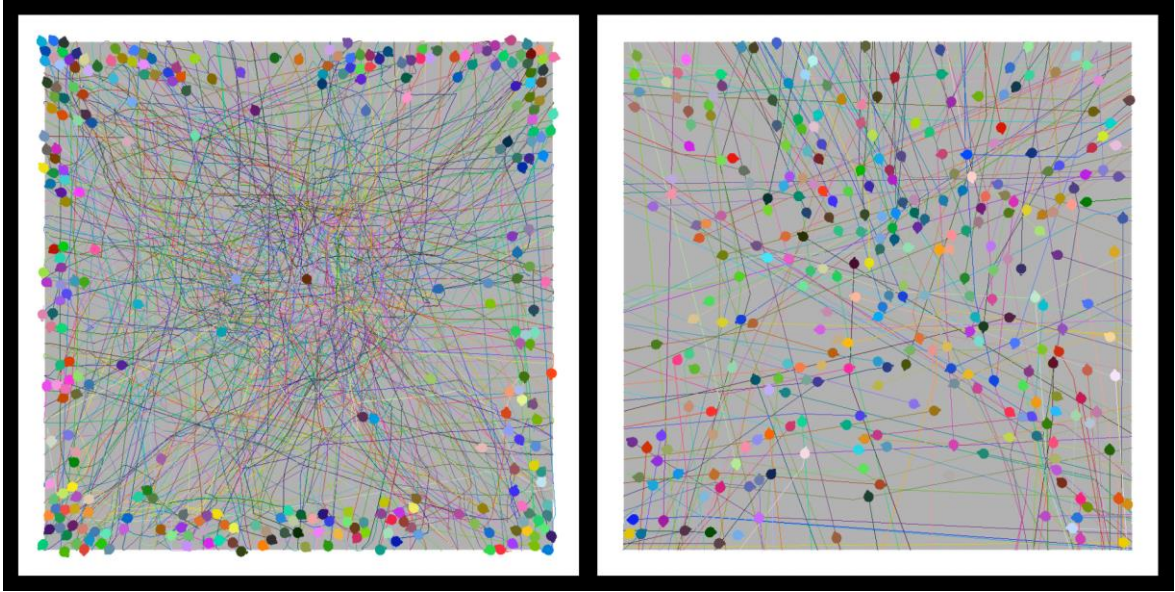
Şekil 6.67 (Sol) RVO ve (Sağ) ÇYPCY yöntemleri ile gerçekleştirilen benzetimlerin 0. saniyesinde yakalanan anlık kuşbakışı görüntüleri



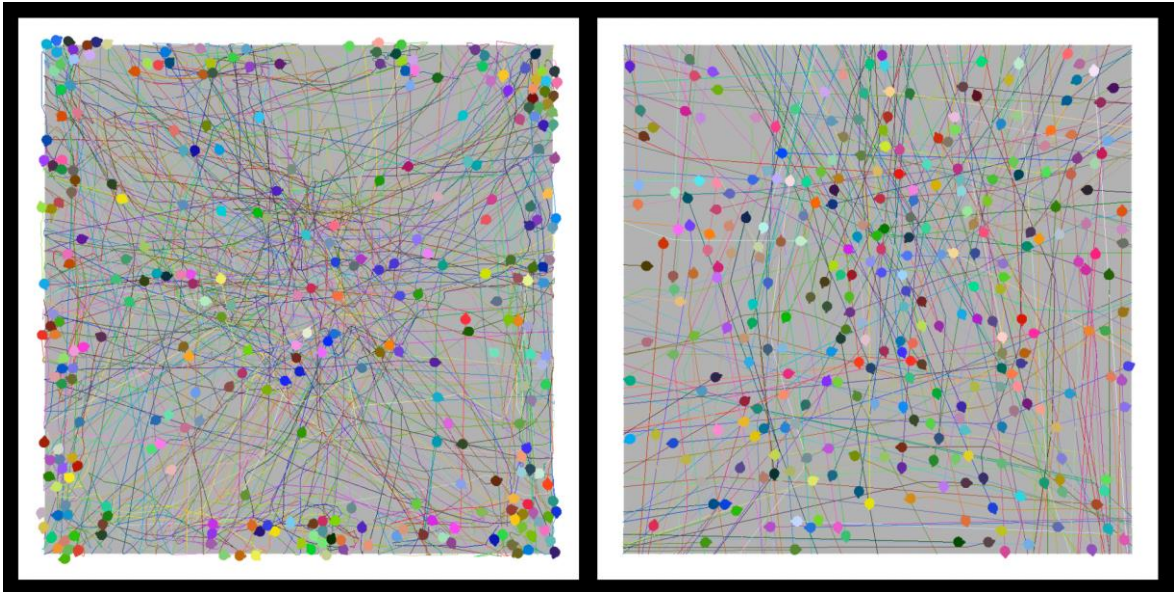
Şekil 6.68 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 10. saniyesinde yakalanan anlık kuşbakışı görüntüleri



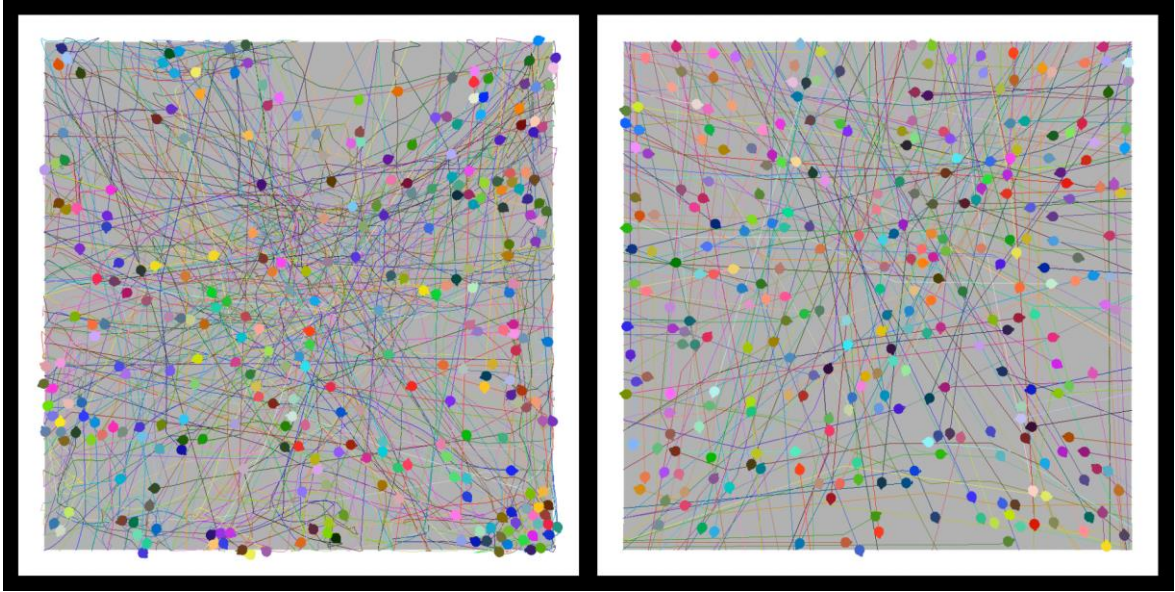
Şekil 6.69 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 20. saniyesinde yakalanan anlık kuşbakışı görüntüleri



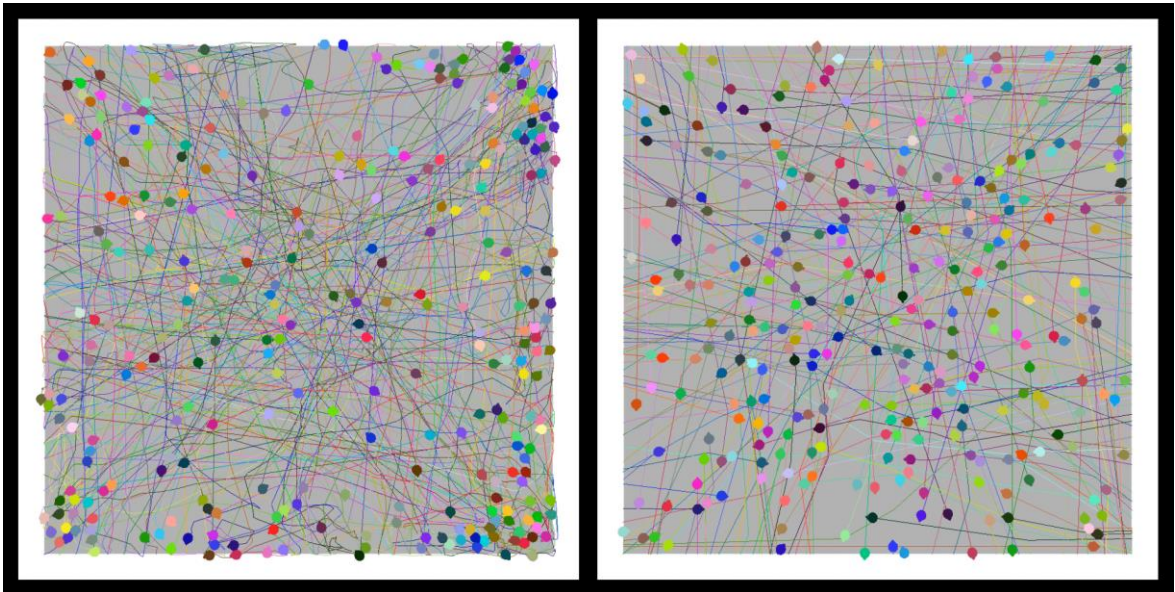
Şekil 6.70 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 30. saniyesinde yakalanan anlık kuşbakışı görüntüleri



Şekil 6.71 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 40. saniyesinde yakalanan anlık kuşbakışı görüntüleri



Şekil 6.72 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 50. saniyesinde yakalanan anlık kuşbakışı görüntüleri



Şekil 6.73 (Sol) RVO ve (Sağ) ÇYPÇY yöntemleri ile gerçekleştirilen benzetimlerin 60. saniyesinde yakalanan anlık kuşbakışı görüntüleri

7. TARTIŞMA VE ÖNERİLER

Kalabalık benzetimlerinde gezinim, temel olarak her bir bireyin bir noktadan başka bir noktaya hareket ettirilmesine dayanır. Ancak bu hareket esnasında var olan diğer bireylerle ve karşılaşılabilir hareketli engellerle yaşanabilecek olası çarpışmaların tespit edilerek bu çarpışmalardan kaçınmayı sağlayacak birtakım manevraların gerçekleştirilmesi gerekir. Söz konusu bu olası çarpışmaların tespit edilmesi başlı başına yeterince karmaşık ve işlem yükü yüksek bir süreç iken her bir birey için bir de bu çarpışmalardan kaçınmayı sağlayacak uygun manevraların hesaplanması gezinim maliyetinin neden bu denli yüksek olduğunu özetlemektedir. Ayrıca her bir bireyin çevresinde çarpışma ihtimali olan birden fazla birey veya engel olması ya da daha uzak gelecekteki çarpışmaları önceden tespit edebilmek ve bunlar için de önlem alabilmek adına her bir birey için kontrol altında tutulan çevrenin daha büyük seçilmesi de gezinim maliyetini katlayan faktörlerdir. Üstüne üstlük gerçek zamanlı benzetimlerde tüm bu işlemlerin her bir birey için her bir zaman adımı içerisinde bir kez yapılma zorunluluğu, problemi daha da içinden çıkılmaz bir hale getirmektedir.

Böylesi bir durumda kullanılabilir işlem gücü, benzetimin kalitesi ve benzetimi yapılabilecek maksimum birey sayısı arasında üç yönlü bir ilişkiden söz etmek mümkündür. Sayılan bu parametrelerin herhangi birinin sabit olduğu kabul edildiğinde diğer ikisi arasında bir denge kurulması gerekmektedir. Bunun en temel nedeni ise işlem gücünün pahalı ve dolayısıyla sınırlı bir kaynak olmasıdır. Nadir olan kimi kullanım senaryolarında karşılık gelen maliyeti göze alarak işlem gücünü artırmak suretiyle istenilen kaliteye ve birey sayısına sahip benzetimler oluşturulması mümkündür. Kimi zaman da gerçekleştirilen akademik araştırmalar veya sektörel gelişmeler neticesinde ortaya çıkan yeni gezinim yöntemleri sayesinde yaklaşık aynı niteliklere sahip bir benzetimin daha düşük işlem maliyeti ile elde edilmesi olanaklı hale gelebilmektedir. Yine de genel uygulama işlem gücünün sabit tutularak diğer değişkenler arasında bir denge kurulması yönündedir. Örneğin, işlem gücünün sabit olduğu bir senaryoda benzetimi yapılacak bireylerin görünüşleri ve hareketleri ne kadar kaliteli ve gerçeğe yakın olursa bu amaçla harcanacak işlem gücü de o denli yüksek olacağından bu gereksinimi karşılamak adına benzetimde yer alabilecek maksimum birey sayısını daha düşük tutmak gerekecektir. Tam tersine sabit işlem gücü ile daha fazla bireyin yer aldığı benzetimler gerçekleştirilmek istendiğinde ise benzetimin kalitesinden ödün vermek kaçınılmaz olacaktır.

Buradan hareketle bu tez çalışması kapsamında 2 boyutlu benzetim alanları üzerinde gerçek zamanlı olarak gerçekleştirilen kalabalık benzetimlerinde yer alan bireylerin gezinim

maliyetini mümkün olan en düşük seviyeye indirmeye çalışırken bu bireylerin gezinim hareketlerinin inandırıcılığını mümkün mertebe korumaya çalışan bir yaklaşım üzerine yoğunlaşmıştır. Bu amaçla yönlendirmesiz gezinim olarak adlandırılan yaklaşımdan faydalanan üç yeni yöntem geliştirilmiştir. Bu yöntemlerde bireylere, benzetime katılmadan hemen önce başlangıç noktalarından bitiş noktalarına kadar uzanan ve çarpışmasız olduğu garanti edilen doğrusal parçalardan oluşan yollar ve bu yol parçaları üzerinde ilerlemelerini sağlayacak sabit yürüme hızları tahsis edilmektedir. Böylelikle benzetimde yer alan hiçbir bireyin, benzetimde kaldığı süre boyunca herhangi bir doğrultu veya hız değişikliği yapmadığı sürece diğer hiçbir birey ile çarpışmayacağı garanti edildiğinden, hiçbir birey için benzetim boyunca herhangi bir anda çarpışma tespiti ve çarpışmadan kaçınma manevrası yapılmasına gerek olmamaktadır. Bu durum, benzetimin gezinim maliyetini mümkün olan en düşük seviyeye, her bir zaman adımında sadece bireylerin pozisyonlarının güncellenmesine, indirgemektedir.

Tez kapsamında geliştirilen yönlendirmesiz gezinim yöntemlerinden ilki tamamıyla AİB üzerinde çalışırken diğer ikisi bireylere atanan çarpışmasız yolların belirlenmesi için GİB'nin özelleşmiş donanımsal görselleştirme özelliklerinden faydalanmaktadır. AİB üzerinde çalışan AİB tabanlı TYPÇY yöntemi yeni oluşturulacak bireyler için TYPÇY'ler belirleyebilmek adına rastgele başlangıç pozisyonları ve hız vektörleri seçerek karşılık gelen yolların benzetimde var olan bireyler ile herhangi bir çarpışma içerip içermediğini AİB üzerinde gerçekleştirdiği testler ile kontrol etmektedir. AİB tabanlı TYPÇY yöntemi, benzetim alanı üzerinde kurulu konum-zamansal bir dizinleme sistemi kullanarak yalnızca aynı zaman aralığında yeni oluşturulacak bireye yakın olacağı için çarpışma ihtimali bulunan bireyler ile çarpışma testleri gerçekleştirilmesini sağlamaktadır. Böylelikle aday yolların her biri için gerçekleştirilecek çarpışma testleri sayısının azaltılmasını ve birey oluşturma maliyetinin en aza indirilmesini hedeflemektedir.

GİB tabanlı diğer iki yöntemde ise yeni bir çarpışmasız yol belirleme sürecinde benzetimde yer alan bireylerin sahip oldukları hız vektörlerine bağlı olarak gezinimleri boyunca 3 boyutlu konum-zamansal uzayda kaplayacakları hacimler (KZYE) GİB ile görselleştirilmiştir. Yine GİB'nin paralel veri işleme yetenekleri kullanılarak konum-zaman uzayında söz konusu bu hacimlerden arta kalan boşluklar değerlendirilerek var olan diğer hiçbir hacim ile kesişmeyecek şekilde benzer yeni bir hacmin tespit edilmesi sağlanmıştır. Belirlenen bu yeni hacim de 2 boyutlu benzetim alanı üzerinde yeni bir çarpışmasız yola ve sabit bir hız vektörüne karşılık geleceğinden bu bilgiler ile yeni bir birey oluşturularak

benzetime dâhil edilmesi sağlanmıştır. GİB tabanlı TYPÇY yönteminde bireylerin başlangıç noktalarından bitiş noktalarına kadar uzanan TYPÇY'ler üzerinde hareket etmeleri sağlanmaktadır. ÇYPÇY yönteminde ise bireylerin başlangıç noktalarından bitiş noktalarına gitmek üzere ÇYPÇY'lere sahip olmalarına olanak verilmiştir. Bu doğrultuda her bir çarpışmasız yolun barındırabileceği parça sayısı için bir üst sınır tanımlanarak bireylerin ihtiyaca göre farklı sayıda parçalardan oluşan çarpışmasız yollar edinmesi sağlanmıştır. ÇYPÇY'ler kullanabilen ÇYPÇY yöntemini, bireylerin sahip olabileceği maksimum yol parçası sayısını bir adet ile sınırlayarak tıpkı TYPÇY yöntemleri gibi çalıştırmak da mümkündür. Ayrıca geliştirilen ÇYPÇY yönteminde bu parçaların birleşim noktalarında ayarlanabilir belirli limitler dâhilinde doğrultu ve yürüme hızı değişiklikleri yapılmasına izin verilmiştir. Bu noktada her üç yöntem için de her bir çarpışmasız yol parçasının sabit bir doğrultu ve sabit bir yürüme hızından meydana gelen sabit bir hız vektörüne karşılık geldiğini hatırlatmakta fayda vardır.

Geliştirilen gezinim yöntemlerini değerlendirmek üzere alanda mevcut en popüler birey tabanlı gezinim yöntemlerinden bir olan RVO metodu ile çeşitli başlıklar altında karşılaştırmalar yapılmıştır. İlk olarak geliştirilen yönlendirmesiz gezinim yöntemlerinin birey oluşturma başarımı incelenmiş ve birbirleri ile kıyaslanmıştır. Bu bağlamda RVO yönteminin hemen hemen maliyetsiz olarak görülebilecek yeni birey oluşturma yaklaşımının, elde edilebilecek birey sayıları üzerinde doğrudan bir etkisi bulunmadığı için RVO ile herhangi bir karşılaştırma yapılmasına gerek duyulmamıştır. Yine de RVO yönteminin deneylerde kullanılan diğer tüm yöntemlerden çok daha yüksek birey sayılarına ulaşmakta zorlanmayacağını belirtmekte fayda vardır. Birey oluşturma başarımını ifade etmek üzere, benzetim boyunca elde edilen ortalama ve maksimum birey sayısı metrikleri kullanılmıştır. GİB tabanlı TYPÇY yönteminin hem ortalama hem de maksimum birey sayıları bakımından AİB tabanlı TYPÇY metoduna göre üstün olduğu görülmüştür. Buna ek olarak, ÇYPÇY yönteminde maksimum parça sayısı bir olarak ayarlanarak AİB ve GİB tabanlı TYPÇY yöntemleriyle doğrudan karşılaştırma sağlanmıştır. Bu şekilde elde edilen benzetimlerin de AİB tabanlı TYPÇY metoduna kıyasla daha yüksek ortalama ve maksimum birey sayılarına ulaşabildiği görülmüştür. Diğer yandan, GİB tabanlı TYPÇY yöntemi ile yalnızca TYPÇY'ler kullanmasına izin verilen ÇYPÇY yönteminin ortalama ve maksimum birey sayılarının karşılıklı olarak birbirine oldukça yakın olduğu belirlenmiştir. Neticede aynı işi farklı şekillerde yapan bu iki yöntemin benzer sonuçlar vermesi gayet normaldir.

Ayrıca her üç yöntemin her bir zaman adımı içerisinde birey oluşturmak için kullanabileceği zaman aralığı ve dolayısıyla gerçekleştirebilecekleri birey oluşturma denemelerinin sayısı artırılarak birey sayılarındaki değişimler gözlemlenmiştir. Ancak her bir zaman adımında gerçekleştirilen birey oluşturma denemesi sayısının artmasına rağmen ne ortalama ne de maksimum birey sayılarının kayda değer miktarda değişmediği sonucuna ulaşılmıştır. Bu yöntemlerin birey sayılarını sınırlayan etken, benzetim alanı üzerinde doğrusal yolları sabit bir hızla takip ederek hareket edebilecek bireylerin sayısının bir üst sınırının yani bir doygunluk noktasının bulunmasıdır. Bu doygunluk noktasına yaklaşıldıkça olası çözümlerin giderek azalması sebebiyle sabit bir hız vektörü ile hareket edebilecek yeni bir birey oluşturmak giderek zorlaşmakta ve doygunluk noktasına gelindiğinde ise ne kadar deneme yapılırsa yapılsın imkânsız hale gelmektedir. AİB tabanlı TYPÇY metodunun birey sayısı doygunluk noktasının, geliştirilen diğer yöntemlere göre bu denli düşük olmasının nedeni ise; bu yöntemde çarpışmasız yolları belirlemek üzere kullanılan başlangıç pozisyonu, bitiş pozisyonu ve yürüme hızı gibi değerlerin tamamen rastgele seçilerek oluşacak sabit hız vektörünün var olan bireyler ile çarpışmaya neden olup olmayacağını test edilmesidir. Çarpışmasız yolları belirlemek üzere kullanılan bu rastgele parametre seçme işlemi nedeniyle olası çözümlerin bazıları göz ardı edildiği için yeni birey oluşturma denemelerinin çoğu var olan bireylerle çarpışmaya yol açarak başarısızlıkla sonuçlanmakta ve aynı özelliklere sahip benzetimlerin birey sayıları da buna bağlı olarak geliştirilen diğer yöntemlerden düşük kalmaktadır.

ÇYPÇY yönteminin ÇYPÇY'ler kullanabilen benzetimlerinde ise kullanılan parametre kombinasyonlarının hemen hemen tamamında yalnızca TYPÇY'ler kullanan tüm yöntemlerden daha yüksek ortalama ve maksimum birey sayılarına ulaşılmıştır. Bu sonucun en temel nedeni, yolların doğrultu ve hız değişiklikleri gösterebilen birden fazla parçadan oluşmasına izin verilmesidir. Bu sayede yeni birey oluşturma denemelerindeki olası çözüm kümesi genişletilerek birey doygunluk noktası yukarı taşınmış ve böylece daha fazla denemenin başarılı sonuçlanarak daha fazla birey oluşturulmasına imkân sağlanmıştır. ÇYPÇY yönteminde kullanılan parametrelerin artan değerlerinin genel olarak elde edilecek birey sayıları üzerinde artışa neden olabileceğini söylemek mümkündür. Yine de olası çözüm kümesi ne kadar geniş olursa olsun tüm bu parametre kombinasyonlarının da birer birey doygunluk sınırı olduğunu belirtmekte fayda vardır.

Daha sonra karşılaştırılan tüm bu yöntemlerin birey oluşturma maliyetlerini değerlendirmek üzere, benzetim boyunca yeni bir birey oluşturmak için harcadıkları süreler ölçülerek

ortalama birey oluşturma süreleri hesaplanmıştır. RVO yöntemini kullanan benzetimler, yeni birey oluşturma sırasında sadece yeni bireyin yerleştirileceği pozisyon nedeniyle o anda benzetimde yer alan yakın bireyler ile çarpışıp çarpışmayacağını kontrol etmektedir. Yeni bireyi içinde bulunduran birkaç birey çiftinin pozisyonları arasındaki uzaklıkların hesaplanmasından ibaret olan bu işlem çok düşük maliyete sahip olduğu için RVO yönteminin birkaç mikro saniye olan ortalama birey oluşturma süresi karşılaştırılan tüm yöntemler arasında açık ara en düşük olanıdır. AİB tabanlı TYPÇY metodu birey oluşturma işlemleri için fazladan birkaç mikro saniye gerektirerek birey oluşturma maliyeti bakımından RVO yönteminin gerisinde yer almaktadır. Bu yöntemin birey oluşturma maliyetinin bu denli yüksek olmasının nedeni, elde edilen rastgele sabit hız vektörleri ile hareket edecek yeni bireylerin var olan diğer bireyler ile çarpışıp çarpışmayacağını test etmek üzere AİB üzerinde birçok işlem gerçekleştirmesidir. GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin sadece TYPÇY'ler kullanan bireylerden oluşan benzetimlerinin ortalama birey oluşturma süreleri ise bir milisaniyenin altındadır. Buna rağmen, GİB tabanlı TYPÇY yönteminde ilk GPGPU geçişinin sonucunda elde edilen atomik sayaç değerine bakılarak birey oluşturma denemesinin başarılı olup olmayacağına göre ikinci GPGPU geçişinin koşullu bir biçimde gerçekleştirilmesine bağlı olarak bu yöntemin ortalama birey oluşturma süresi ÇYPCY yöntemininkinden bir miktar daha düşüktür. Ancak GİB tabanlı TYPÇY ve ÇYPCY yöntemlerinin yeni birey oluşturma sürecindeki adımların tamamının işletildiği başarılı birey oluşturma denemeleri göz önünde bulundurulduğunda, GİB tabanlı TYPÇY yönteminin ortalama birey oluşturma maliyeti, olası tüm çözümleri transfer etmesinin gerektirdiği ek süre ihtiyacından kaynaklı olarak ÇYPCY yöntemine göre biraz daha fazladır. Yine de GİB tabanlı TYPÇY ve ÇYPCY metotları özellikle daha yavaş bir aktarım olduğu bilinen GİB belleğinden AİB belleğine veri transferi gerçekleştirdiği için AİB ile GİB arasında hiçbir yönde veri transferi gerektirmeyen AİB tabanlı TYPÇY metodundan daha yüksek birey oluşturma sürelerine ihtiyaç duymaktadır. ÇYPCY yönteminin ÇYPCY'ler kullanan benzetimlerinde kullanılan parametre kombinasyonlarına bağlı olarak bir milisaniyenin üzerine çıkabilen ortalama birey oluşturma süreleri her durumda geliştirilen üç yöntem de dâhil olmak üzere TYPÇY'ler kullanan tüm yöntemlerin birey oluşturma maliyetlerinden daha yüksektir. Bu durumun ana nedeni, birey oluşturma maliyetinde en büyük paya çarpışmasız parçaların belirlenmesi işlemlerinin sahip olması ve bireylerin kullandığı ortalama parça sayısının artmasına bağlı olarak ortalama birey oluşturma sürelerinin de yükselmesidir. Yapılan istatistiksel analizler sonucunda çarpışmasız yollar için izin verilen

maksimum parça sayısının çok yüksek etki büyüklüğüne sahip olması da birey oluşturma maliyeti üzerindeki bu etkisini doğrular niteliktedir.

Son olarak, geliştirilen yönlendirmesiz gezinim yöntemleri ile önceki deneylerde kullanılan RVO yönteminin gezinim maliyetleri kıyaslanmıştır. Bu amaçla benzetim boyunca birey başına harcanan gezinim süreleri ölçülerek birey başına ihtiyaç duyulan ortalama gezinim süreleri hesaplanmıştır. RVO yöntemi her bir zaman adımında olası çarpışmalardan kaçınmak adına her bir bireyin hız vektörünü güncellemek üzere maliyetli işlemler gerçekleştirdiği için birey başına ortalama gezinim maliyeti bir mikro saniyenin üzerinde olup tüm yöntemler içerisinde açık ara en yüksek gezinim maliyetine sahip yöntemdir. AİB tabanlı TYPÇY metodunun gezinim için birey başına harcadığı ortalama süre GİB tabanlı TYPÇY ve ÇYPÇY yöntemleri ile gerçekleştirilen tüm benzetimlerin birey başına gezinim maliyetinden daha yüksektir. Bu durumun nedeni, AİB tabanlı TYPÇY metodunun birey oluşturma sürecinde kullanılan konum-zamansal veri yapısının her bir zaman adımında bireylerin hareketlerine göre güncellenmesi gerekliliğinin, bireylerin gezinim maliyetine bir ek olarak yansımadır. GİB tabanlı TYPÇY ve ÇYPÇY yöntemleriyle yalnızca TYPÇY'ler kullanarak gerçekleştirilen benzetimlerin gezinim maliyetleri arasında önemli bir fark bulunmamakla birlikte bu maliyetler tüm yöntemler ile gerçekleştirilen tüm benzetimler arasında elde edilen en düşük gezinim maliyetlerine karşılık gelmektedir. Diğer taraftan, ÇYPÇY yönteminin ÇYPÇY'ler kullanan benzetimlerinde parçalar arası geçişlerde gerçekleştirilen hız vektörü ve benzeri değişkenlerin güncellemesi işlemlerine bağlı olarak bu benzetimlerde gezinim için birey başına harcanan ortalama süreler, GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin TYPÇY'ler kullanan benzetimlerine kıyasla bir miktar daha yüksektir. Geziniim maliyetlerindeki bu artış kullanılan parametre kombinasyonlarına bağlı olarak değişmektedir. AİB tabanlı TYPÇY metodu da dâhil olmak üzere deneylerde kullanılan tüm yönlendirmesiz gezinim yöntemlerinin gezinim maliyetlerinin RVO yönteminden çok daha düşük olması, tezin en önemli motivasyonu olan gerçek zamanlı benzetimlerde düşük maliyetli gezinim hedefine ulaşıldığının en büyük göstergesidir. Ayrıca geliştirilen yönlendirmesiz gezinim yöntemlerinde birey başına gezinim maliyetinin birey sayısındaki artıştan etkilenmemesi de belirtilmesi gereken bir diğer önemli ayrıntıdır.

Özetlemek gerekirse tez kapsamında 2 boyutlu benzetim alanları üzerinde gerçekleştirilecek kalabalık benzetimlerinde bireylerin gezinimini sağlamak üzere geliştirilen üç yeni yöntem AİB üzerindeki gezinim maliyeti yükünü önemli ölçüde azaltmaktadır. Bunu yaparken AİB üzerindeki gezinim hesaplamalarının bir kısmını veya tamamını GİB üzerine kaydırarak

AİB'nin işlem gücünden tasarruf etmek yerine, yönlendirmesiz gezinim yaklaşımını kullanıp bireylerin olası çarpışmaları tespiti ve bu çarpışmalardan kaçınmaları adına gerçekleştirilen işlemlere olan ihtiyacı ortadan kaldırarak gezinim hesaplamalarını minimum düzeye indirmektedir. Geliştirilen bu yönlendirmesiz gezinim yöntemlerinden ilki, bireylerin manevra yapmaksızın gezinimini sağlayan çarpışmasız yolların belirlenmesini sağlamak üzere yalnızca AİB üzerinde çalışmaktadır. Diğer ikisi ise çarpışmasız yol belirleme sürecinde GİB'den faydalanmaktadır. Kalabalık benzetimlerinde çoğu zaman sadece görselleştirme amacıyla kullanılan GİB, buradaki kullanımıyla ekstra bir işlem gücü olarak görev yapmaktadır. Dolayısıyla potansiyel kullanıcıların geliştirilen bu yöntemlerin sadece AİB değil GİB üzerine yüklediği maliyetleri de göz önünde bulundurarak karar vermeleri tavsiye edilmektedir. Bu doğrultuda ihtiyaç duyulan benzetimlerin çalışacağı donanım üzerinde yer alan GİB'nin sahip olduğu işlem gücünün kullanım oranını değerlendirmek yerinde olacaktır. Eğer AİB'nin işlem gücünde tasarrufa ihtiyaç varken GİB'nin işlem kapasitesinin görece az bir bölümü kullanılıyorsa, GİB'ye ek yük bindirecek GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin kullanılmasının önünde bir engel bulunmamaktadır. Tam aksine eğer GİB görselleştirme veya farklı ihtiyaçlar için tam kapasiteye yakın çalıştırılıyorsa bu durumda GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin kullanımı GİB üzerinde darboğaz oluşturarak gerçek zamanlı benzetimlerde “saniye başına resim karesi” (*frame per second*) olarak bilinen değer (SBRK) düşmesine neden olacaktır. Diğer taraftan, GİB bulundurmeyen donanımlar üzerinde gerçekleştirilecek benzetimler için ise GİB tabanlı TYPÇY ve ÇYPÇY yöntemlerinin kullanılması mümkün olmayacaktır. Bu gibi durumlarda benzetimin birey sayısı ihtiyaçlarının karşılanması koşuluyla tez kapsamında geliştirilen AİB tabanlı TYPÇY yönteminin kullanımı gündeme gelmekle birlikte ihtiyaca yönelik alternatif çözümlere yönelmek de değerlendirilebilecek bir diğer seçenektir.

Geliştirilen yönlendirmesiz gezinim yöntemlerinden hangisinin kullanılacağıyla ilgili seçim yapma konusunda, gezinim maliyetine ek olarak benzetimlerde ihtiyaç duyulan birey sayısının yanı sıra bu bireylerin gezinim hareketlerinin kalitesi de göz önünde bulundurulmalıdır. Birey doygunluğu sınırı ve hiç manevra içermemesi hedeflenen benzetimin ihtiyaçlarını karşılıyor ise geliştirilen gezinim yöntemlerinin her üçünün de yalnızca TYPÇY'ler kullanan benzetimlerinden faydalanılabilir. Yine de yaklaşık gezinim maliyetlerine sahip olmalarına rağmen ÇYPÇY yöntemine kıyasla ortalamada daha düşük birey oluşturma maliyetine sahip GİB tabanlı TYPÇY yönteminin kullanılmasının GİB

üzerine nispeten daha az yük bindireceğini hatırlatmakta fayda vardır. Diğer taraftan, AİB tabanlı TYPÇY yönteminin bu iki yöntemden de çok daha düşük birey oluşturma maliyetine sahip olduğunu ve buna ek olarak, GİB üzerine hiç yük bindirmeyeceğini belirtmek gerekmektedir. Eğer bireylerin gezinim hareketlerinin daha kaliteli olması isteniyor ise aynı birey sayılarına çok rahat ulaşabilecek ÇYPÇY yönteminin ÇYPÇY'ler kullanan benzetimlerinden de yararlanmak mümkündür. Ancak buradan TYPÇY'ler üzerinde sabit bir hızla hareket etmek kalitesiz bir gezinim oluşturur gibi yargı çıkartılmamalıdır. Nitekim yörünge kalitelerinin değerlendirildiği deneylerde de görüldüğü üzere, her bir zaman adımında olası çarpışmalardan kaçınmak adına bireylerin hız vektörlerini güncelleyen birey tabanlı yaklaşımların pek çoğunda, ardışık güncellemeler arasındaki aşırı farklılıklardan kaynaklanan tereddüt hareketi benzeri salınımlar da gerçekçilik algısını zedelemektedir. Çünkü daha önce de belirtildiği üzere canlıların hedeflerine ulaşmak üzere mümkün olan en az enerjiyi harcayacakları yolu seçecekleri [10] bilindiğinden bu gibi manevralar gereksiz olarak nitelendirilerek inandırıcılığı olumsuz yönde etkilemektedir. Böyle gereksiz manevralar yapmaksızın dosdoğru hedefine yol alan bireylerin oluşturduğu benzetimlerin yeterince ve hatta daha inandırıcı olabileceğinin düşünülmesine neden olan da bahsi geçen mümkün olan en az enerji prensibidir. ÇYPÇY yöntemiyle gerçekleştirilen benzetimlerde parçaların birleşim noktalarında yapılan hız vektörü değişiklikleri ise yakın gelecekteki çarpışmalardan kaçınmak adına gerekli olduğu için benzetimin inandırıcılığını olumlu yönde etkileyeceği yönünde bir varsayım oluşturulmuştur.

Eğer hedeflenen benzetimlerin birey sayısı ihtiyacı yalnızca TYPÇY'ler kullanan benzetimlerin doygunluk sınırının üzerinde yer alıyorsa bu durumda sadece ÇYPÇY yönteminin kullanımı seçenek olarak yer alacaktır. ÇYPÇY yöntemi kullanılırken seçilecek farklı parametre kombinasyonları ile yaklaşık aynı birey sayılarına ulaşmanın mümkün olduğu, gerçekleştirilen deneylerde de görülmüştür. Bu noktada kullanıcıların parametre değerlerini seçerken ihtiyaç duyulan birey sayılarına ulaşabilen kombinasyonlar arasında birey oluşturma maliyeti ve birey gezinim maliyeti en düşük olanları tercih etmeleri tavsiye edilmektedir. Parametrelerin artan değerleriyle birlikte genel olarak birey sayılarının yanı sıra birey oluşturma ve az da olsa birey gezinim maliyetlerinin de artacağı akılda bulundurulduğunda genel bir yaklaşım olarak yaklaşık aynı birey sayılarına ulaşabilen en düşük parametre değerlerine sahip kombinasyonun seçilmesi yerinde olacaktır. Bunun dışında, ÇYPÇY yönteminin temel farklılığını oluşturduğu için nispeten daha özel bir parametre olan maksimum parça sayısının üzerinde hassasiyetle durulması gerekmektedir.

Birey sayısını istenilen seviyeye çıkarmak üzere doğrudan maksimum parça sayısını artırmak, birey sayısının yanı sıra birey oluşturma maliyetini de aynı ve belki de daha da yüksek oranda artıracaktır. Çünkü bu parametrenin birey oluşturma maliyeti üzerindeki etkisi, birey sayısı üzerindeki etkisinden daha büyüktür. Bu nedenle birey sayısını artırmak üzere çok yüksek sayıda parçaya izin vermek yerine parça sayısını daha makul bir seviyede seçip, olası çözüm uzayını genişletirken maliyeti daha az etkileyen doğrultu ve hız değişikliği gibi diğer parametrelerin değer aralığını artırmak, birey oluşturma maliyetini düşük tutmak adına daha anlamlı olacaktır.

Diğer taraftan, geliştirilen gezinim yöntemlerinin maliyet testleri çevrimdışı olarak gerçekleştirildiği için her bir zaman adımında yeni birey oluşturma denemeleri için ayrılan süre parametresinin etkileri doğrudan incelenememiştir. Ancak açıktır ki her bir zaman adımında herhangi bir iş için ayrılan sabit bir süreyi artırmak, zaman adımlarının tamamlanma süresini de aynı oranda uzatarak bir saniye içerisinde gerçekleştirilebilecek zaman adımı sayısını ve dolayısıyla SBRK'yi düşürerek benzetimin akıcılığını olumsuz etkileyecektir. Birey sayılarının incelendiği deneylerde her bir zaman adımında birey oluşturma denemeleri için ayrılan sürenin en küçük istatistiksel etki büyüklüğüne sahip olduğu göz önünde bulundurulduğunda bu sürenin artırılmasının birey sayıları üzerinde oldukça sınırlı bir artışa neden olacağı sonucuna ulaşılmaktadır. Bu durumun ana nedeni zaman adımları içerisinde gerçekleştirilen yeni birey oluşturma denemelerinin sayısının artırılmasına rağmen bu denemelerin başarılı olanlarının sayısında önemli bir artış olmamasından dolayı yeni oluşturulan birey sayısının çok fazla değişmemesidir. Bu nedenle benzetimlerin akıcılığını korumak adına ihtiyaç duyulan benzetimlerde birey sayısı gereksinimlerini karşılamak üzere her bir zaman adımında birey oluşturma için ayrılan sürenin artırılması başvurulacak son tercih olarak değerlendirilmelidir.

Geliştirilen yönlendirmesiz gezinim yöntemlerinde benzetim alanı üzerinde yer alan sabit engellere yer verilmesi de mümkündür. Söz konusu bu sabit engeller, beklemekte olan durağan bir birey olabileceği gibi hareketsiz bir nesne veya yapı da olabilir. Sabit engeller içeren benzetimlerin birey oluşturma başarımını incelemek adına farklı sayılarda daire ve kare biçiminde engeller içeren kare ve dikdörtgen benzetim alanları üzerinde ortalama ve maksimum birey sayılarının ölçüldüğü benzetimler gerçekleştirilmiştir. Sabit engeller içermesi nedeniyle yeni birey oluşturmak üzere olası çözümlerin sayısının azaldığı bu benzetimlerdeki ortalama ve maksimum birey sayıları, engel içermeyen aynı büyüklük ve şekildeki alanlar üzerinde gerçekleştirilen benzetimlere kıyasla beklenileceği üzere bir

miktar daha düşüktür. Buna karşın, ÇYPÇY yöntemiyle sabit engeller içeren alanlar üzerinde gerçekleştirilen benzetimler, GİB tabanlı TYPÇY yöntemiyle birey doldurulmaya çalışılan aynı büyüklük ve şekildeki alanların hem engel içeren hem de engel içermeyen türevlerinden daha yüksek ortalama ve maksimum birey sayılarına ulaşabilmektedir. Buradan hareketle ÇYPÇY yönteminin birden çok yörünge parçası kullanması ve bu parçalar arasındaki geçişlerde sınırlı da olsa doğrultu ve hız değişikliklerine izin vermesinin sabit engeller içeren benzetim alanları üzerinde (konum-zamansal uzaylar içerisinde) bile TYPÇY yöntemlerine göre olası çözüm kümesini genişleterek daha fazla birey oluşturulmasını olanaklı hale getirdiği söylenebilir. Bu sayede, sabit engeller nedeniyle TYPÇY'ler kullanan bireylerin geziniminin mümkün olmadığı benzetim alanları üzerinde ÇYPÇY yöntemi ile gezinim sağlanabilir. Her ne kadar sabit engeller içeren daha karmaşık benzetim senaryoları ile bunların birey sayısı ve maliyet araştırmaları tezin odağını dağıtmamak adına kapsam dışında tutulmuş olsa da gelecekte araştırılması muhtemel konulardan birini oluşturmaktadır.

Değinilmesi gereken bir diğer önemli ayrıntı ise geliştirilen gezinim yöntemlerinin, benzetim boyunca benzetim alanının özelliklerinin yanı sıra bireylerin karar verilen çarpışmasız yollarının değişmeyeceğini varsaymasıdır. Benzetim alanının şekil veya boyutlarının değişmesi, benzetim alanı üzerindeki sabit engellere yenilerinin eklenmesi, var olan engellerin konum veya kapladığı alanların değişmesi; benzetimde var olan bireylerin mevcut yollarında öngörülemeyen çarpışmalara neden olabileceği gibi değişen koşullara göre yeni oluşturulacak bireylere, var olan bireyler ya da engellere çarpmalarına neden olabilecek yollar atanmasına da yol açabilir. Benzer biçimde, benzetimde var olan bireylerin önceden planlanan çarpışmasız yollara göre hareket etmeyip konum veya hız vektörlerinin değişmesi de diğer bireylerle ve engellerle çarpışmalarıyla sonuçlanabilir. Bu nedenle geliştirilen yönlendirmesiz gezinim yöntemlerinin yalnızca, benzetimlerde yer alan bu unsurların konum, şekil, büyüklük, hız vektörü gibi niteliklerinin kullanıcıyla ve birbirleriyle etkileşimi sonucu değişmeyeceği senaryolar için kullanımının uygun olduğunu hatırlatmak yerinde olacaktır.

Geliştirilen bu yöntemlerin etkileşimsiz ambiyans kalabalıkları oluşturmak amacıyla kullanılmak üzere önerilmesinin temel nedenlerinden biri bu iken bir diğeri de sınırlı gezinim hareketi kabiliyetine sahip bireylerin kullanıcıların algı odağından uzak tutulmak istenmesidir. Ancak yörünge kalitesini değerlendirmek adına yapılan deneyler kapsamında sınırsız manevra kabiliyetine sahip RVO yöntemiyle gerçekleştirilen benzetimlerde gereksiz

salınım manevraları, karşı yönden gelen bireyler nedeniyle istemeden yapılan geri dönüş ve sürüklenmelerin yanı sıra birey yörüngelerinin aynı zaman diliminde kesiştiği dar bölgelerde takılmalar görülmektedir. Geliştirilen gezinim yöntemleri ise mümkün olan en az enerji prensibine göre hedeflerine doğru en kısa yolu izlemek üzere minimum sayıda manevra yapacağı için sadece RVO yöntemine değil diğer birey tabanlı gezinim yöntemlerinin de birçoğuna kıyasla çok daha tutarlı ve pürüzsüz yörüngeler üzerinde hareket eden bireyler oluşturmaktadır. Bu nedenle her ne kadar geliştirilen yönlendirmesiz gezinim yöntemleriyle gerçekleştirilen benzetimlerin çok daha gerçekçi olacağı düşünülse de bu varsayımı desteklemek üzere insan deneklerin kullanıcı olarak yer alacağı kapsamlı algı deneyleri düzenlenmesi gerekmektedir. İhtiyaç duyulan bu deneylerde incelenmesi gereken parametre sayısı burada sunulan yöntemler içerisinde yer alanlar ile bir araya geldiğinde başlı başına farklı ve büyük bir başka araştırmanın konusu olabileceğinden söz konusu bu deneyler tez çalışması kapsamı dâhiline alınmayarak gelecekte üzerinde çalışılması planlanan olası araştırma alanları arasında bırakılmıştır.

Tez kapsamında geliştirilen yönlendirmesiz gezinim yöntemleri, kalabalık parçaları yaklaşımının türevleri içerisinde TYPÇY veya ÇYPCY'lerden oluşan kalabalık parçalarının gerçek zamanlı belirlenmesi için kullanılabilir. Bu parçalar üzerindeki giriş-çıkış pozisyonları, yanlarına ekleneceği diğer parçalarla uyumluluk gösterecek biçimde seçileceğinden yeni oluşturulacak parçalar üzerindeki çarpışmasız yolların rastgelelikten tamamen kurtularak çok daha belirleyici bir biçimde tespit edilmesi sağlanabilir. Böylelikle kalabalık parçaları yaklaşımlarında ihtiyaç duyulan tüm parçaların önceden oluşturularak yüksek bellek gerektirecek biçimde saklanması önüne geçilebilir. Benzer biçimde, sınırlı sayıda hız vektörü değişikliği manevrası gerçekleştiren bireyler oluşturan bu yönlendirmesiz gezinim yöntemleri, önceden kaydedilmiş gezinim betiklerinin otomatik bir biçimde oluşturulması esnasında kullanılarak, betikler içerisinde yer alacak hız vektörü değişikliği kayıtları sayısı ve buna bağlı olarak da yüksek bellek maliyeti önemli ölçüde azaltılabilir.

Diğer taraftan, geliştirilen yönlendirmesiz gezinim yöntemlerinin, yürüyen bireylerden meydana gelen kalabalıklar oluşturmak dışında trafikte hareket eden araçların benzetimini yapmak üzere kullanılması mümkündür. Özellikle yolların doğrusal olarak ifade edilebildiği senaryolarda, her bir yolu sadece karşılıklı iki kenarı arasında çift yönlü bir hareket akışına izin verilen dikdörtgen biçiminde bir benzetim alanı olarak varsaymak ve yol parçaları olarak düşünülebilecek bu benzetim alanlarının birleşim noktalarında, ÇYPCY yönteminde olduğu gibi doğrultu ve hız değişikliklerine izin vermek, düşük gezinim maliyetine sahip

trafik benzetimi gerçekleştirmek amacıyla kullanılabilir bir çözümdür. Diğer taraftan, geliştirilen yönlendirmesiz gezinim yöntemlerinin robotik alanında, düzlemsel bir ortam üzerinde hareket eden otonom robotlar geliştirmek amacıyla kullanılması da olanaklıdır. Bu bağlamda kullanılacak robotların her biri üzerinde basit de olsa ayrı bir GİB donanımı bulunması sağlanarak robotların gezinim için kullanacakları çarpışmasız yolların dağıtık bir biçimde hesaplanması da sağlanabilir. Ancak bunun için her bir robot, üzerinde gezindiği alanın yanı sıra diğer robotların anlık konum ve hız gibi bilgilerine de ihtiyaç duyacağından bu bilgilerin gerçek zamanlı paylaşımını sağlayacak bir sisteme ihtiyaç olduğunu da belirtmekte fayda vardır.

Bunların dışında geliştirilen gezinim yöntemlerinde kullanılan GİB tabanlı çarpışmasız, doğrusal ve sabit hızlı konum-zamansal yol belirleme yaklaşımı, temelde 3 boyutlu sınırlı dikdörtgen prizma biçiminde hacme sahip bir uzay içerisinde yer alan hiçbir engele çarpmadan, bu uzayın bir yüzeyinden diğer bir yüzeyine doğrusal bağlantılar oluşturacak tek bir doğru parçası ya da ardışık uç uca eklenmiş doğru parçalarının gerçek zamanda belirlenmesi probleminin özelleştirilmiş bir çözümdür. Tez kapsamında geliştirilen bu özel çözümden uzayda yer alan engeller, bireylere karşılık gelen karmaşık 3 boyutlu cisimleri içine alan basitleştirilmiş sınırlayıcı hacimler olarak kullanılmış olmakla birlikte problemin ihtiyacına göre doğrudan bu engellere karşılık gelen 3 boyutlu modellerin görselleştirilmesiyle daha duyarlı çözümler üretilmesi de mümkündür. AİB üzerinde özellikle tüm bu karmaşık 3 boyutlu şekillere sahip engelleri göz önünde bulundurarak iki farklı yüzey arasında bağlantı oluşturacak olası doğruların yalnızca bir tanesinin belirlenmesi bile oldukça yüksek işlem gücü gerektiren hesaplamalarla dolu karmaşık algoritmalara ihtiyaç duymaktadır. Oysaki GİB aracılığıyla uzay sınırlarını belirleyen herhangi bir yüzey üzerinde seçilecek bir noktaya konumlandırılmış perspektif bakışa sahip bir kamera kullanılarak uzaydaki bu engellerin görselleştirilmesi sonucu, tek bir seferde ve zahmetsiz bir biçimde diğer bir yüzeyle bağlantı oluşturacak tüm olası doğruların belirlenebileceği görülmüştür. Dolayısıyla tez kapsamında geliştirilen bu çözümün bilgisayar bilimlerinde veya farklı bir alanda benzer bir probleme uyarlanarak ihtiyaç duyulan doğru parçası ya da ardışık bağlantılı doğru parçalarının GİB yardımıyla gerçek zamanlı olarak hızlı ve kolay bir biçimde belirlenmesi de bilimsel veya akademik kullanım alanlarının genel ifadesini teşkil etmektedir.

8. SONUÇ

Bu tez kapsamında yönlendirmesiz gezinim olarak adlandırılan yaklaşımı kullanan üç yeni yöntem geliştirilmiştir. Bu yöntemlerde kullanılan yönlendirmesiz gezinim yaklaşımı, bireylere benzetime katılmadan hemen önce başlangıç noktalarından bitiş noktalarına kadar uzanan ve bir veya daha fazla doğrusal, sabit hızlı yörünge parçasından oluşan çarpışmasız yollar tahsis edilmesi ve bireylerin çarpışmasız olduğu garanti edilen bu yollar üzerinde manevra yapmaksızın ilerlemeleri esasına dayanmaktadır. Böylelikle hiçbir birey için benzetimin herhangi bir anında çarpışma tespiti ve çarpışmadan kaçınma manevrası yapılmasına gerek olmayacağından benzetimin gezinim maliyeti mümkün olan en düşük seviyeye, her bir zaman adımında yalnızca bireylerin pozisyonlarının güncellenmesine, indirgenmiştir. Bu nedenle geliştirilen üç yöntemin de birey başına gezinim maliyetleri, alanda mevcut güncel birey tabanlı bir gezinim yöntemine kıyasla çok daha düşüktür.

Geliştirilen ilk gezinim yönteminde AİB üzerinde rastgele seçilen parametrelere göre belirlenen ve yalnızca tek bir yörünge parçasından oluşabilen çarpışmasız yollar, ikinci gezinim yönteminde GİB tabanlı daha belirleyici bir yaklaşım ile oluşturulmuştur. Üçüncü gezinim yönteminde GİB tabanlı bu çarpışmasız yol belirleme yaklaşımının daha da geliştirilmesiyle bireylerin birden çok yörünge parçasının uç uca eklenmesiyle oluşturulan ve bu yörünge parçaları arasındaki geçişlerde belirli limitler dâhilinde doğrultu ve hız değişiklikleri içerebilen yollara sahip olmalarının önü açılmıştır. Çarpışmasız yolların belirlenmesi sürecinin giderek iyileştirilmesine paralel olarak geliştirilen üç farklı yönlendirmesiz gezinim yönteminde, artan birey oluşturma maliyetine rağmen benzetimi yapılabilecek birey sayılarının yanı sıra bu bireylerin hareket çeşitliliği de giderek artmaktadır. Özellikle son gezinim yöntemiyle kazanılan bu sınırlı, ancak etkili manevra kabiliyeti sabit engeller içeren benzetim alanları üzerinde gerçekleştirilen benzetimlerde ulaşılabilecek birey sayısı yönünden başarıyı artırmıştır. Bununla birlikte bireylerin yörünge parçaları arasındaki geçiş noktalarında, çarpışmadan kaçınma manevraları olarak algılanabilecek davranışlar sergilemesini sağlayarak benzetimlerin inandırıcılığını da olumlu yönde etkilemiştir. Ayrıca geliştirilen bu yöntemlerde yer alan bireylerin alanda var olan birey tabanlı yöntemlerin birçoğunun aksine gereksiz veya hatalı hiçbir manevra yapmayarak tutarlı ve pürüzsüz bir biçimde hareket etmeleri de benzetimi yapılan kalabalıkların gerçekçilik algısını artıran bir diğer önemli unsurdur.

Geliştirilen yönlendirmesiz gezinim yöntemlerinin temel motivasyonu, minimum gezinim maliyetine sahip etkileşimsiz ambiyans kalabalıkları oluşturmaktır. Buna ek olarak, bu

yöntemlerin; her bir şeridi tek yönlü hareket akışına sahip trafik benzetimleri gerçekleştirmek, düzlemsel hareket eden otonom robotların gezinimi için gerek duyulan çarpışmasız yolların dağıtık bir biçimde hesaplanmasını sağlamak, kalabalık parçaları yöntemlerinde ihtiyaç duyulan kalabalık parçalarını ve bunlar üzerindeki çarpışmasız yörüngeleri gerçek zamanlı bir biçimde oluşturarak bu parçaların saklanması için kullanılan bellek ihtiyacını azaltmak ya da önceden kaydedilmiş gezinim betiklerini düşük bellek maliyetiyle otomatik olarak oluşturmak üzere kullanılması da mümkündür.

KAYNAKLAR

- [1] Dobbyn, S., Hamill, J., O’Conor, K., O’Sullivan, C., Geopostors: A Real-Time Geometry / Impostor Crowd Rendering System, *Proceedings of the 2005 symposium on Interactive 3D graphics and games - SI3D ’05*, 95, 1, **2005**.
- [2] Ahn, J., Oh, S., Wohn, K., Optimized motion simplification for crowd animation, *Computer Animation and Virtual Worlds*, 17, 155–165, **2006**.
- [3] Kulpa, R., Olivierxs, A.-H., Ondřej, J., Pettré, J., Imperceptible relaxation of collision avoidance constraints in virtual crowds, *ACM Transactions on Graphics*, 30, 1, **2011**.
- [4] Paris, S., Gerdelan, A., O’Sullivan, C., CA-LOD: Collision Avoidance Level of Detail for Scalable, Controllable Crowds, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5884 LNCS, 13–28, **2009**.
- [5] Toledo, L., De Gyves, O., Rudomín, I., Hierarchical level of detail for varied animated crowds, *The Visual Computer*, 30, 949–961, **2014**.
- [6] Yersin, B., Ma"im, J., Pettré, J., Thalmann, D., Crowd Patches: Populating Large-Scale Virtual Environments for Real-Time Applications, *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2009*, 207–214, **2009**.
- [7] Ramirez, J. G. R., Lange, D., Charalambous, P., Esteves, C., Pettré, J., Optimization-based computation of locomotion trajectories for crowd patches, *Proceedings of the Seventh International Conference on Motion in Games - MIG ’14*, 7–16, **2014**.
- [8] Jordao, K., Charalambous, P., Christie, M., Pettré, J., Cani, M.-P., Crowd art: Density and flow based crowd motion design, *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG 2015*, **2015**.
- [9] Yotsumoto, Y., Kuwahara, M., Sato, T., Wriggling Motion Trajectory Illusion, *Journal of Vision*, 12, 1234–1234, **2012**.
- [10] Zipf, G. K., *Human Behaviour and the Principle of Least Effort*, Addison-Wesley Press, **1949**.
- [11] Inman, V. T., Ralston, H. J., Todd, F., Lieberman, J. C., *Human walking*, Williams & Wilkins, **1981**.
- [12] Still, G. K., *Crowd dynamics*, University of Warwick, **2000**.
- [13] Guy, S., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D., Pedestrians: a least-effort approach to crowd simulation, *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 119, **2010**.
- [14] Pettré, J., Lin, M., New generation crowd simulation algorithms, *ACM SIGGRAPH 2014 Courses on - SIGGRAPH ’14*, 1–72, **2014**.
- [15] Xu, M.-L., Jiang, H., Jin, X.-G., Deng, Z., Crowd Simulation and Its Applications: Recent Advances, *Journal of Computer Science and Technology*, 29, 799–811, **2014**.
- [16] Zhou, S. vd., Crowd modeling and simulation technologies, *ACM Transactions on Modeling and Computer Simulation*, 20, 1–35, **2010**.

- [17] Peters, C., Ennis, C., McDonnell, R., O’Sullivan, C., Crowds in context: Evaluating the perceptual plausibility of pedestrian orientations, *Eurographics 2008*, **2008**.
- [18] Barut, O., Haciomeroglu, M., Ozcan, C. Y., Sever, H., A Dynamic Exit Choice Method for Real-Time Indoor Evacuation Scenarios, *Proceedings of Eurasia Graphics*, **2014**.
- [19] O’Connor, S., Liarokapis, F., Peters, C., A perceptual study into the behaviour of autonomous agents within a virtual urban environment, *2013 IEEE 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2013*, **2013**.
- [20] Hawe, G. I., Coates, G., Wilson, D. T., Crouch, R. S., Agent-based simulation for large-scale emergency response, *ACM Computing Surveys*, 45, 1–51, **2012**.
- [21] Thompson, P. A., Marchant, E. W., A computer model for the evacuation of large building populations, *Fire Safety Journal*, 24, 131–148, **1995**.
- [22] He, G., Yang, Y., Chen, Z., Gu, C., Pan, Z., A review of behavior mechanisms and crowd evacuation animation in emergency exercises, *Journal of Zhejiang University SCIENCE C*, 14, 477–485, **2013**.
- [23] Helbing, D., Farkas, I., Vicsek, T., Simulating dynamical features of escape panic, *Nature*, 407, 487–490, **2000**.
- [24] Bohannon, J., BUILDING SAFETY: Directing the Herd: Crowds and the Science of Evacuation, *Science*, 310, 219–221, **2005**.
- [25] Schelhorn, T., O’Sullivan, D., Haklay, M., Thurstain-goodwin, M., Streets: An Agent-based Pedestrian Model, CUPUM’99 Computers in urban planning and urban management on the edge of the millenium: Proceedings of the 6th International Conference. FrancoAngeli: Venice, Italy., 14, **1999**.
- [26] Penn, A., Turner, A., Space syntax based agent simulation, *1st International Conference on Pedestrian and Evacuation Dynamics*, 99–114, **2001**.
- [27] Turner, A., Penn, A., Encoding Natural Movement as an Agent-Based System: An Investigation into Human Pedestrian Behaviour in the Built Environment, *Environment and Planning B: Planning and Design*, 29, 473–490, **2002**.
- [28] Moussaid, M., Helbing, D., Theraulaz, G., How simple rules determine pedestrian behavior and crowd disasters, *Proceedings of the National Academy of Sciences*, 108, 6884–6888, **2011**.
- [29] Molnar, P., Starke, J., Control of distributed autonomous robotic systems using principles of pattern formation in nature and pedestrian behavior, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 31, 433–435, **2001**.
- [30] McPhail, C., Powers, W. T., Tucker, C. W., Simulating Individual and Collective Action in Temporary Gatherings, *Social Science Computer Review*, 10, 1–28, **1992**.
- [31] TUCKER, C. W., SCHWEINGRUBER, D., MCPHAIL, C., Simulating arcs and rings in gatherings, *International Journal of Human-Computer Studies*, 50, 581–588, **1999**.
- [32] Lin, M. C., Manocha, D., Virtual cityscapes: recent advances in crowd modeling and traffic simulation, *Frontiers of Computer Science in China*, 4, 405–416, **2010**.

- [33] Sewall, J., Wilkie, D., Merrell, P., Lin, M. C., Continuum Traffic Simulation, *Computer Graphics Forum*, 29, 439–448, **2010**.
- [34] Lin, M. C. vd., Interactive Modeling, Simulation and Control of Large-Scale Crowds and Traffic, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5884 LNCS, 94–103, **2009**.
- [35] Shen, J., Jin, X., Detailed traffic animation for urban road networks, *Graphical Models*, 74, 265–282, **2012**.
- [36] Wilkie, D., Sewall, J., Lin, M. C., Transforming GIS Data into Functional Road Models for Large-Scale Traffic Simulation, *IEEE Transactions on Visualization and Computer Graphics*, 18, 890–901, **2012**.
- [37] Sewall, J., van den Berg, J., Lin, M. C., Manocha, D., Virtualized Traffic: Reconstructing Traffic Flows from Discrete Spatiotemporal Data, *IEEE Transactions on Visualization and Computer Graphics*, 17, 26–37, **2011**.
- [38] Sewall, J., Wilkie, D., Lin, M. C., Interactive hybrid simulation of large-scale traffic, *ACM Transactions on Graphics*, 30, 1, **2011**.
- [39] HENDERSON, L. F., The Statistics of Crowd Fluids, *Nature*, 229, 381–383, **1971**.
- [40] Wu, J., Popović, Z., Realistic modeling of bird flight animations, *ACM Transactions on Graphics*, 22, 888, **2003**.
- [41] Klotsman, M., Tal, A., Animation of Flocks Flying in Line Formations, *Artificial Life*, 18, 91–105, **2011**.
- [42] Wang, X., Jin, X., Deng, Z., Zhou, L., Inherent noise-aware insect swarm simulation, *Computer Graphics Forum*, 33, 51–62, **2014**.
- [43] Tu, X., Terzopoulos, D., Artificial Fishes: Physics, Locomotion, Perception, Behavior, *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '94*, 43–50, **1994**.
- [44] Massive Software, 2017, <http://www.massivesoftware.com/> (May15, **2017**).
- [45] Golaem Crowd, 2017, <http://golaem.com/> (May15, **2017**).
- [46] Thalmann, D., O’Sullivan, C., Ciechomski, P. de H., Dobbyn, S., EG 2006 Course on Populating Virtual Environments with Crowds, *Eurographics 2006: Tutorials*, **2006**.
- [47] Azahar, M. A. B. M., Sunar, M. S., Daman, D., Bade, A., Survey on Real-Time Crowds Simulation, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5093 LNCS, 573–580, **2008**.
- [48] Kapadia, M., Shoulson, A., Boatright, C. D., Huang, P., Durupinar, F., Badler, N. I., What’s Next? The New Era of Autonomous Virtual Humans, *Motion in Games*, 170–181, **2012**.
- [49] Magnenat-Thalmann, N., Thalmann, D., Virtual humans: thirty years of research, what next?, *The Visual Computer*, 21, 997–1015, **2005**.
- [50] Kapadia, M., Badler, N. I., Navigation and steering for autonomous virtual humans, *Wiley Interdisciplinary Reviews: Cognitive Science*, 4, 263–272, **2013**.

- [51] Ijaz, K., Sohail, S., Hashish, S., A Survey of Latest Approaches for Crowd Simulation and Modeling using Hybrid Techniques, *17th UKSIM-AMSS International Conference on Modelling and Simulation*, 111–116, **2015**.
- [52] Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., Xu, L.-Q., Crowd analysis: a survey, *Machine Vision and Applications*, 19, 345–357, **2008**.
- [53] Thalmann, D., Grillon, H., Maim, J., Yersin, B., Challenges in Crowd Simulation, *2009 International Conference on CyberWorlds*, 1–12, **2009**.
- [54] Helbing, D., Twining, C. J., Bulsari, A., Oliker, S., Furst, M., Maimon, O., A Fluid-Dynamic Model for the Movement of Pedestrians, *Complex Systems*, 6, 391–415, **1993**.
- [55] Hughes, R. L., A continuum theory for the flow of pedestrians, *Transportation Research Part B: Methodological*, 36, 507–535, **2002**.
- [56] Hughes, R. L., The Flow of Human Crowds, *Annual Review of Fluid Mechanics*, 35, 169–182, **2003**.
- [57] Treuille, A., Cooper, S., Popović, Z., Continuum crowds, *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, 1160, **2006**.
- [58] Reynolds, C. W., Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH Computer Graphics*, 21, 25–34, **1987**.
- [59] Reynolds, C. W., Steering behaviors for autonomous characters, *Game Developers Conference*, 370, 763–782, **1999**.
- [60] Helbing, D., Molnár, P., Social force model for pedestrian dynamics, *Physical Review E*, 51, 4282–4286, **1995**.
- [61] Helbing, D., Molnar, P., Self-Organization Phenomena in Pedestrian Crowds, *Environment and Planning B: Planning and Design*, 28, 361–383, **1998**.
- [62] Yu, W. J., Chen, R., Dong, L. Y., Dai, S. Q., Centrifugal force model for pedestrian dynamics, *Physical Review E*, 72, 26112, **2005**.
- [63] Fiorini, P., Shiller, Z., Motion Planning in Dynamic Environments Using Velocity Obstacles, *The International Journal of Robotics Research*, 17, 760–772, **1998**.
- [64] van den Berg, J., Ming Lin, Manocha, D., Reciprocal Velocity Obstacles for real-time multi-agent navigation, *2008 IEEE International Conference on Robotics and Automation*, 1928–1935, **2008**.
- [65] Guy, S. J. vd., ClearPath: Highly Parallel Collision Avoidance for Multi-Agent Simulation, *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09*, 177, 1, **2009**.
- [66] Blue, V., Adler, J., Emergent Fundamental Pedestrian Flows from Cellular Automata Microsimulation, *Transportation Research Record: Journal of the Transportation Research Board*, 1644, 29–36, **1998**.
- [67] AEA Technology, AEA Technology, A Technical Summary of the AEA EGRESS Code, **2002**.
- [68] Kretz, T., Schreckenberg, M., F.A.S.T. - Floor field- and Agent-based Simulation Tool, *International Symposium of Transport Simulation (ISTS06)*, **2006**.

- [69] Anh, N. T. N., Daniel, Z. J., Du, N. H., Drogoul, A., An, V. D., A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7068 LNAI, 371–383, **2012**.
- [70] Wei, X., Xiong, M., Zhang, X., Chen, D., A Hybrid Simulation of Large Crowd Evacuation, *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, 971–975, **2011**.
- [71] Xiong, M., Lees, M., Cai, W., Zhou, S., Low, M. Y. H., Hybrid modelling of crowd simulation, *Procedia Computer Science*, 1, 57–65, **2010**.
- [72] Narain, R., Golas, A., Curtis, S., Lin, M. C., Aggregate dynamics for dense crowd simulation, *ACM Transactions on Graphics*, 28, 1, **2009**.
- [73] Haciomeroglu, M., Barut, O., Ozcan, C. Y., Sever, H., A GPU-assisted hybrid model for real-time crowd simulations, *Computers & Graphics*, 37, 862–872, **2013**.
- [74] Xiong, M., Tang, S., Zhao, D., A Hybrid Model for Simulating Crowd Evacuation, *New Generation Computing*, 31, 211–235, **2013**.
- [75] Patil, S., van den Berg, J., Curtis, S., Lin, M. C., Manocha, D., Directing Crowd Simulations Using Navigation Fields, *IEEE Transactions on Visualization and Computer Graphics*, 17, 244–254, **2011**.
- [76] Tissera, P. C., Printista, A. M., Luque, E., A Hybrid Simulation Model to Test Behaviour Designs in an Emergency Evacuation, *Procedia Computer Science*, 9, 266–275, **2012**.
- [77] Haciomeroglu, M., Ozcan, C. Y., Barut, O., Seckin, L., Sever, H., Hardware-accelerated dynamic clustering of virtual crowd members, *Computer Animation and Virtual Worlds*, 24, 143–153, **2013**.
- [78] Ryder, G., Day, A. M., Survey of Real-Time Rendering Techniques for Crowds, *Computer Graphics Forum*, 24, 203–215, **2005**.
- [79] Beacco, A., Pelechano, N., Andújar, C., A Survey of Real-Time Crowd Rendering, *Computer Graphics Forum*, 35, 32–50, **2016**.
- [80] Clark, J. H., Hierarchical geometric models for visible surface algorithms, *Communications of the ACM*, 19, 547–554, **1976**.
- [81] Garland, M., Heckbert, P. S., Surface simplification using quadric error metrics, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, 209–216, **1997**.
- [82] Hoppe, H., Progressive meshes, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, 99–108, **1996**.
- [83] Hoppe, H., View-dependent refinement of progressive meshes, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, 189–198, **1997**.
- [84] Hoppe, H., Efficient implementation of progressive meshes, *Computers & Graphics*, 22, 27–36, **1998**.
- [85] Sander, P. V., Snyder, J., Gortler, S. J., Hoppe, H., Texture mapping progressive meshes, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, 409–416, **2001**.

- [86] Shiue, L., Goel, V., Peters, J., Mesh Mutation in Programmable Graphics Hardware, *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, 15–24, **2003**.
- [87] Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B., Huebner, R., *Level of Detail for 3D Graphics*, **2003**.
- [88] Loop, C., Smooth Subdivision Surfaces Based on Triangles, *Department of Mathematics, The University of Utah, Masters Thesis*. 74, **1987**.
- [89] Zorin, D., Schröder, P., Sweldens, W., Interpolating Subdivision for meshes with arbitrary topology, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, 189–192, **1996**.
- [90] Kahler, K., Haber, J., Seidel, H.-P., Dynamic refinement of deformable triangle meshes for rendering, *Proceedings. Computer Graphics International 2001*, 285–290, **2001**.
- [91] Vlachos, A., Peters, J., Boyd, C., Mitchell, J. L., Curved PN triangles, *Proceedings of the 2001 symposium on Interactive 3D graphics - SI3D '01*, 159–166, **2001**.
- [92] Tecchia, F., Chrysanthou, Y., Real-Time Rendering of Densely Populated Urban Environments, *Rendering Techniques*, 83–88, **2000**.
- [93] Tecchia, F., Loscos, C., Chrysanthou, Y., Visualizing Crowds in Real-Time, *Computer Graphics Forum*, 21, 753–765, **2002**.
- [94] Aubel, A., Boulic, R., Thalmann, D., Real-time display of virtual humans: levels of details and impostors, *IEEE Transactions on Circuits and Systems for Video Technology*, 10, 207–217, **2000**.
- [95] Kavan, L., Dobbyn, S., Collins, S., Žára, J., O’Sullivan, C., Polypostors: 2D Polygonal Impostors for 3D Crowds, *Proceedings of the 2008 symposium on Interactive 3D graphics and games - SI3D '08*, 149, **2008**.
- [96] Yee, J., Davis, J., Crowd Rendering with Non-Planar 3D Impostors, 59593, **2008**.
- [97] Beacco, A., Spanlang, B., Andújar, C., Pelechano, N., Output-sensitive rendering of detailed animated characters for crowd simulation, *Actas del XX Congreso Espa`nol de Informática Gráfica*, 37–46, **2010**.
- [98] Yuksel, K. A., Yucebilgin, A., Balcisoy, S., Ercil, A., Real-time feature-based image morphing for memory-efficient impostor rendering and animation on GPU, *The Visual Computer*, 29, 131–140, **2013**.
- [99] Savoye, Y., Meyer, A., Multi-layer level of detail for character animation, *5th Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2008*, 57–66, **2008**.
- [100] Rodriguez, R., Cerezo, E., Baldassarri, S., Seron, F. J., New approaches to culling and LOD methods for scenes with multiple virtual actors, *Computers & Graphics*, 34, 729–741, **2010**.
- [101] Gröschel, A., Towards Believable Crowd Simulation for Interactive Real-Time Applications, Hochschule für Technik und Wirtschaft Berlin, **2011**.
- [102] Yilmaz, E., *Massive Crowd Simulation with Parallel Processing*, Middle East Technical University, **2010**.

- [103] McDonnell, R., Dobbyn, S., O’Sullivan, C., LOD Human Representations: A Comparative Study, *Proceedings of the International Workshop on Crowd Simulation*, 101–115, **2005**.
- [104] Hamill, J., McDonnell, R., Dobbyn, S., O’Sullivan, C., Perceptual Evaluation of Impostor Representations for Virtual Humans and Buildings, *Computer Graphics Forum*, 24, 623–633, **2005**.
- [105] McDonnell, R., Dobbyn, S., Collins, S., O’Sullivan, C., Perceptual evaluation of LOD clothing for virtual humans, *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 117–126, **2006**.
- [106] McDonnell, R., Newell, F., O’Sullivan, C., Smooth movers: perceptually guided human motion simulation, *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 1, 259–269, **2007**.
- [107] McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., O’Sullivan, C., Clone attack! Perception of crowd variety, *ACM Transactions on Graphics*, 27, 1, **2008**.
- [108] McDonnell, R., Ennis, C., Dobbyn, S., O’Sullivan, C., Talking Bodies: Sensitivity to Desynchronization of Conversations, *ACM Transactions on Applied Perception*, 6, 1–8, **2009**.
- [109] McDonnell, R., Jörg, S., Hodgins, J. K., Newell, F., O’Sullivan, C., Evaluating the effect of motion and body shape on the perceived sex of virtual characters, *ACM Transactions on Applied Perception*, 5, 1–14, **2009**.
- [110] McDonnell, R., Jörg, S., McHugh, J., Newell, F. N., O’Sullivan, C., Investigating the role of body shape on the perception of emotion, *ACM Transactions on Applied Perception*, 6, 1–11, **2009**.
- [111] McDonnell, R., Larkin, M., Hernández, B., Rudomin, I., O’Sullivan, C., Eye-catching crowds: saliency based selective variation, *ACM Transactions on Graphics*, 28, 1, **2009**.
- [112] Peters, C., Ennis, C., Modeling groups of plausible virtual pedestrians, *IEEE Computer Graphics and Applications*, 29, 54–63, **2009**.
- [113] McHugh, J. E., McDonnell, R., O’Sullivan, C., Newell, F. N., Perceiving emotion in crowds: The role of dynamic body postures on the perception of emotion in crowded scenes, *Experimental Brain Research*, 204, 361–372, **2010**.
- [114] Ennis, C., Peters, C., O’Sullivan, C., Perceptual effects of scene context and viewpoint for virtual pedestrian crowds, *ACM Transactions on Applied Perception*, 8, 1–22, **2011**.
- [115] Hoyet, L., McDonnell, R., O’Sullivan, C., Push it Real: Perceiving Causality in Virtual Interactions, *ACM Transactions on Graphics*, 31, 1–9, **2012**.
- [116] McDonnell, R., Breidt, M., Bühlhoff, H. H., Render me Real? Investigating the Effect of Render Style on the Perception of Animated Virtual Humans, *ACM Transactions on Graphics*, 31, 1–11, **2012**.
- [117] Jarabo, A., Eyck, T. Van, Sundstedt, V., Bala, K., Gutierrez, D., O’Sullivan, C., Crowd Light: Evaluating the Perceived Fidelity of Illuminated Dynamic Scenes, *Computer Graphics Forum*, 31, 565–574, **2012**.
- [118] Amanatides, J., Woo, A., A Fast Voxel Traversal Algorithm for Ray Tracing, *Eurographics*, 87, 3–10, **1987**.

- [119] Perspective Frustum, 2017, <http://www.lighthouse3d.com/tutorials/view-frustum-culling/> (Mayıs, **2017**).
- [120] Perspective Frustum Planes, 2017, <http://www.lighthouse3d.com/tutorials/view-frustum-culling/view-frustums-shape/> (Mayıs, **2017**).
- [121] Perspective Frustum View, 2017, <https://www.script-tutorials.com/webgl-with-three-js-lesson-9/> (Mayıs, **2017**).
- [122] Cormen, T. H., Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms, Second Edition*, 7, **2001**.
- [123] Rosenfeld, A., Connectivity in Digital Pictures, *Journal of the ACM*, 17, 146–160, **1970**.
- [124] Di Stefano, L., Bulgarelli, A., A simple and efficient connected components labeling algorithm, *Proceedings 10th International Conference on Image Analysis and Processing*, 322–327, **1999**.
- [125] Suzuki, K., Horiba, I., Sugie, N., Linear-time connected-component labeling based on sequential local operations, *Computer Vision and Image Understanding*, 89, 1–23, **2003**.
- [126] Dillencourt, M. B., Samet, H., Tamminen, M., A general approach to connected-component labeling for arbitrary image representations, *Journal of the ACM*, 39, 253–280, **1992**.
- [127] Wu, K., Otoo, E., Suzuki, K., Optimizing two-pass connected-component labeling algorithms, *Pattern Analysis and Applications*, 12, 117–135, **2009**.
- [128] Haralick, R. M., Some Neighborhood Operators, *Real-Time Parallel Computing*, (ed.: Onoe, M., Preston, K., ve Rosenfeld, A.), Springer US, Boston, MA, 11–35, **1981**.
- [129] Lumia, R., Shapiro, L., Zuniga, O., A new connected components algorithm for virtual memory computers, *Computer Vision, Graphics and Image Processing*, 22, 287–300, **1983**.
- [130] Fu Chang, Chun-Jen Chen, A linear-time component-labeling algorithm using contour tracing technique, *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 741–745, 1, **2003**.
- [131] Hawick, K. A., Leist, A., Playne, D. P., Parallel graph component labelling with GPUs and CUDA, *Parallel Computing*, 36, 655–678, **2010**.
- [132] Št'ava, O., Beneš, B., Connected Component Labeling in CUDA, *GPU Computing Gems Emerald Edition*, Elsevier, 569–581, **2011**.
- [133] Soh, Y., Ashraf, H., Hae, Y., Kim, I., A Hybrid Approach to Parallel Connected Component Labeling Using CUDA, *International Journal of Signal Processing Systems*, 1, 130–135, **2013**.
- [134] Atomic Counter, 2017, https://www.khronos.org/opengl/wiki/Atomic_Counter (Mayıs, **2017**).
- [135] Blending, 2017, <https://www.khronos.org/opengl/wiki/Blending> (Mayıs, **2017**).
- [136] Depth Test, 2017, https://www.khronos.org/opengl/wiki/Depth_Test (Mayıs, **2017**).

- [137] Ondřej, J., Pettré, J., Olivier, A.-H., Donikian, S., A synthetic-vision based steering approach for crowd simulation, *ACM SIGGRAPH 2010 papers on - SIGGRAPH '10*, 1, 29, **2010**.
- [138] Dutra, T. B., Marques, R., Cavalcante-Neto, J. B., Vidal, C. A., Pettré, J., Gradient-based steering for vision-based crowd simulation algorithms, *Computer Graphics Forum*, 36, 337–348, **2017**.
- [139] Fruin, J. J., *Pedestrian Planning and Design*, Revised Ed, **1987**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Öner BARUT
Doğum Yeri : Ankara
Medeni Hali : Bekâr
E-posta : barutoner@gmail.com
Adresi : Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü,
06800, Beytepe, Ankara, Türkiye

Eğitim

Lise : Mustafa Kemal Yabancı Dil Ağırlıklı Lisesi, Ankara, Türkiye
(2001-2005)
Lisans : Hacettepe Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye
(2006-2010)
Yüksek Lisans : -
Doktora : Hacettepe Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı, Ankara, Türkiye
(2010-2017)

Yabancı Dil ve Düzeyi

İngilizce : YÖKDİL - 91.25 (09.07.2017)

İş Deneyimi

Araştırma Görevlisi : Hacettepe Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü (2011 - Halen)

Deneyim Alanları

Bilgisayar Grafiği, Kalabalık Benzetimi, GPGPU, Doğal Dil İşleme, Makine Öğrenmesi

Tezden Üretilmiş Projeler ve Bütçeleri

-

Tezden Üretilmiş Yayınlar

- O. Barut, M. Hacımeroglu, “Real-time collision-free linear trajectory generation on GPU for crowd simulations”, The Visual Computer, vol. 31, no. 6–8, pp. 843–852, Jun. **2015**.
- O. Barut, M. Hacımeroglu, E. Sezer, “Combining GPU-generated Linear Trajectory Segments to Create Collision-free Paths for Real-time Ambient Crowds”, Graphical Models, (büyük düzeltme).

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

- O. Barut, M. Hacımeroglu, C. Y. Ozcan, “Illusive Crowd”, in 27th International Conference on Computer Animation and Social Agents (CASA 2014), Teksas, ABD, **2014**.



HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 26/09/2017

Tez Başlığı / Konusu: **İnandırıcı Kalabalık Benzetiminde Birey Sayısı ve Hareket Çeşitliliği Araştırması**

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam **218** sayfalık kısmına ilişkin, **26/09/2017** tarihinde şahsım/tez danışmanım tarafından **Turnitin** adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % **1** 'dir.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

26/09/2017

Adı Soyadı: Öner BARUT
Öğrenci No: N10161987
Anabilim Dalı: Bilgisayar Mühendisliği
Programı: Bilgisayar Mühendisliği
Statüsü: Y.Lisans Doktora Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Prof. Dr. Ebru AKÇAPINAR SEZER