

T.C Kütahya Sağlık Bilimleri Üniversitesi

Motion Capture 3D: Gerçek Zamanlı Hareket Yakalama

Ders Sorumlusu: Emre Güngör

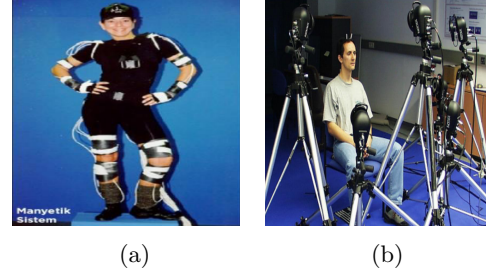
İlknur Koparır 2118121010

22.04.2024

1 Giriş

Hareket yakalama teknolojisi (motion capture), yüzyılı aşkın bir süre içerisinde insanlığın geliştirdiği teknolojinin etkileri ile büyük ölçüde gelişim göstermiştir. Günümüzde sanat, eğlence ve teknoloji dünyasında önemli bir yere sahip olan motion capture, insanların gerçek dünyadaki hareketleri dijital ortama aktarma ihtiyacından doğmuştur. Daha hızlı ve daha ucuz bilgisayar donanımının, daha ucuz ve daha iyi dijital kameralarla birleşmesiyle, video tabanlı uygulamalar giderek daha yaygın hale gelmiştir. Özellikle sinema, video oyunları, reklam endüstrisi gibi alanlarda, gerçekçi hareketlerin oluşturulması ve karakterlerin doğal bir şekilde davranması büyük bir talep görmesinden dolayı bu teknolojilerin kullanım alanları gelişmektedir. Hareket halindeki bir obje veya bir canlının üç boyutlu koordinat ve açısal değişimlerinin çeşitli yöntemlerle bir alıcıya iletilmesi olarak tanımlanan hareket yakalama teknolojisi [1], işaretleyici tabanlı ve işaretleyici tabanlı olmayan sistemler olarak ikiye ayrılmıştır. İşaret tabanlı hareket yakalama sistemleri manyetik veya optik vericilerin insan vücudunun önemli eklem bölgelerine yerleştirilerek hareket bilgisinin üretilmesidir. İşaret tabanlı olmayan hareket tanıma sistemlerde ise verici doğrudan figür olarak karşımıza çıkmaktadır. Kameralar, insan formunu arka plandan ayırtan algoritma ve derin öğrenme teknikleri sayesinde verilerden hareket bilgisi çıkarılmaktadır. Bu teknolojiler sayesinde daha hızlı

ve hassas hareket yakalama sistemleri geliştirilerek, gerçek zamanlı etkileşimli uygulamaların performansı artırılmakta ve kullanıcı deneyimi iyileştirilmektedir.



Şekil 1: (a) Manyetik hareket yakalama sistemi, (b) İşaretleyici tabanlı olmayan hareket yakalama sistemi.

2 Literatür Taraması

Motion capture teknolojisinin kökenleri, eski çağlardan başlayarak, insanların hareketlerini taklit etme ve canlandırma ihtiyacıyla başlamıştır.

1906'da Amerikalı mucit Thomas Edison'un yardımı ile Amerikan gazetesi karikatüristi James Stuart Blanton'un, Şekil 2'de görüldüğü üzere ilk çizgi filmi olan Humorous Phases of Faces'i (Komik Yüzlerin Mizahi Evreleri) gösterime girmiştir. Çizgi filmde, kara tahta üzerinde, bir şapka ile oynayan

palyaço ve çemberden atlayan bir köpek de dahil olmak üzere, elle çizilmiş karakterler bulunmaktaydı. Hareket ettirip durdurma tekniği kullanılan filmde, çizimler kendi başlarına tamamlanıyor ve sonra hareket eden çizimler gibi görünmekteydi. Bu teknik izleyiciler tarafından komik bulunmuş ve ilgi ile karşılanmıştır [2]



Şekil 2: Humorous Phases Faces

1959'da Harrison, potansiyometrelerle (ayarlanabilir dirençler) donatılmış bir elbiseyi oyuncuya giydirmiş ve oyuncunun tüm hareketlerini bir ekran yardımı ile gerçek zamanlı olarak kaydetmeyi başarmıştır. Bu ilkel bir teçhizatı ancak gerçek zamanlı hareket yakalamasının ilk örneğidir. 1980'lere gelindiğinde animatörler, oyuncuların hareketlerini izlemek için aktif işaretçilerle kaplı vücut giysileri ve avuç büyüklüğünde kameralar kullanmışlardır [2].

Teknolojinin gelişmesiyle birlikte Rosales ve ark. [3], kalibre edilmemiş birden fazla görüş kullanarak 3B eklemli duruşun (pose) tahmini için bir çerçeve sunar. Bu çerçevede, Uzmanlaştırılmış Haritalama Mimarisi (SMA) adı verilen istatistiksel bir çıkarım yöntemi kullanılır. SMA, kareler ve görünümler arasında 2B eklem konumları sağlar.

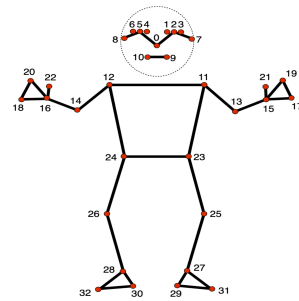
Kakadiaris ve Metaxas [4], dik açılı konfigürasyonlarda çoklu kamera görüntülerini kullanarak insan vücut parçalarının 3B model tabanlı izleme ve şekil tahmini için bir yöntem sunarlar. Daha sonra yaklaşımı, çoklu görüşlü video dizilerinden insan hareketinin tahmini ve buna göre animasyon dizileri oluşturmak için genişlettiler.

3 Metodoloji

İnsan vücut hareketlerini kaydetmek ve dijital ortama aktarmak, günümüzde animasyon endüstrisinde ve interaktif teknolojilerde önemli bir rol oynamaktadır.

3.1 Mediapipe

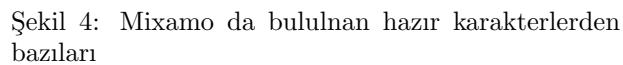
Mediapipe, Google tarafından geliştirilmiş ücretsiz ve açık kaynaklı bir makine öğrenmesi kütüphanesidir.[5] Medya akışı (video, ses) üzerinde gerçek zamanlı veri işleme, nesne tespiti, nesne takibi, el ve yüz algılama gibi çeşitli görevleri gerçekleştirmek için tasarlanmıştır.[6] MediaPipe'nin Holistic yöntemi, aynı anda yüz, el ve vücut pozisyonlarının takibini mümkün kılmaktadır. Bu yöntem, Şekil 3'de görülebileceği üzere tek bir kare üzerinden 33 noktanın tespit edilmesine izin verirken, geleneksel COCO topolojisinde tek bir kare üzerinde 17 noktanın tespit edilmesini mümkün kılmaktadır. Bu yöntemle, 33 poz, 21 el ve 468 yüz işaret noktası olmak üzere toplamda 543 işaret noktası tespit edilebilir. Proje kapsamında OpenCV ve MediaPipe kütüphaneleri birlikte kullanılarak kamera üzerinde insan vücudunun tespiti yapılmıştır. Bu yöntemle, kameradan alınan görüntüler gerçek zamanlı olarak işlenerek insan vücudunun çeşitli parçaları, yani yüz, el ve vücut pozisyonları tespit edilmiştir. MediaPipe ile tespit edilen vücut parçalarının x, y, z koordinatları daha sonra kullanılmak üzere kaydedilmiştir.



Şekil 3: MediaPipe'da 33 vücut parçasının konumu [7]

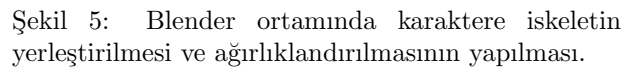
Mixamo, Adobe tarafından sunulan bir hizmet olup, sanat projelerinde, filmlerde ve oyunlarda kullanılmak üzere herkesin erişebileceği çevrim içi bir karakter ve mocap animasyon veritabanıdır.[8] Blender ise açık kaynak kodlu bir üç boyutlu modelleme ve animasyon yazılımıdır. Blender ile animasyon, görsel efekt, üç boyutlu model ve sanal gerçeklik modelleri üretiminde kullanılmaktadır.[9]

Eğer projeleriniz için özel bir karakteriniz yoksa Mixamo kullanmamız için önceden hazırlanmış karakterler sunmaktadır. Karakter seçmek için ekranın üst kısmındaki “Karakterler” sekmesine tıklamamamız gerekmektedir. Karakter seçimi yaptıktan sonra karakteri fbx formatında indirmemiz gerekmektedir.



3.2.2 Blender ortamında karaktere iskelet yerleştirilmesi

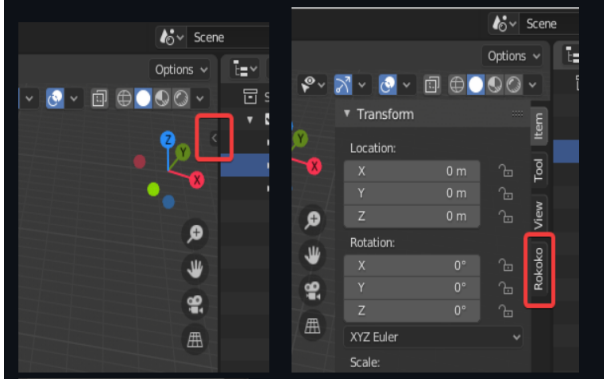
yapısına uygun olarak düzenlenerek eklem bölgeleri doğru şekilde konumlandırılmıştır.



Rokoko Studio eklentisi, Blender’da animasyon oluşturmak için kullanılan bir araçtır.Bu eklenti sayesinde, hareket yakalama verileri dijital karaktere kolaylıkla aktarılabilir.[6]

İlk olarak, Rokoko'nun web sitesine[11] giderek Blender için uygun olanı dosyayı indirmemiz gerekmektedir. Sonrasında, Blender'ı açıp, 'Edit' menüsünden 'Preferences' bölümüne giriyoruz. Ardından, 'Add-ons' sekmesine geçip, indirdiğimiz zip dosyasını Blender'a eklemek için 'Install' düğmesine tıklıyoruz.

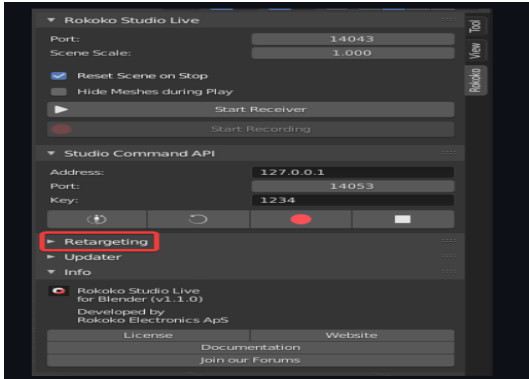
- 3



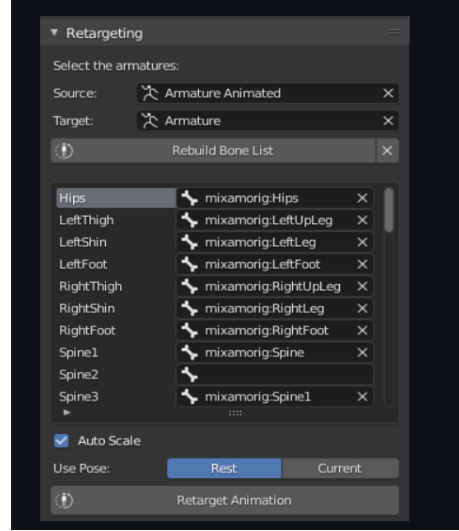
Şekil 6: Blender ortamında Rokoko kurulumu.

3.3.2 Karaktere hareket verilmesi

İlk olarak, Blender'da Retargeting panelini açmamız gerekmektedir. Bu panel, animasyonların yeniden hedeflenmesi işlemlerini gerçekleştirmemize olanak sağlar. İkinci adımda, animasyonun kaynak iskeletini ve animasyonun hedef alınacağı iskeleti seçmemiz gerekmektedir. Kaynak iskelet, animasyonun geldiği orijinal iskeleti temsil ederken, hedef iskelet, animasyonun yeniden hedefleneceği iskeleti temsil eder. Bu adımda, Blender'ın bu iki iskelet arasındaki kemik listesini oluşturması için "Build Bone List" düğmesine basmamız gerekecek. Blender'ın oluşturduğu kemik listesini kontrol etmek gerekmektedir. Eğer eksik veya yanlış kemikler varsa, bunların düzeltilmesi yapılır. Ardından animasyona hareket verilir.



Şekil 7: Retargeting.



Şekil 8: İki iskelet arasında kemiklerin eşleştirilmesi.



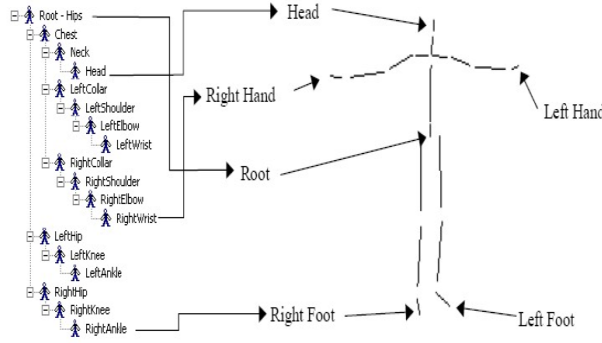
Şekil 9: Rokoko eklentisi kullanılarak animasyona hareket verme.

3.4 Bvh Dosya Formatı

BVH dosya formatının geçmişi ve gelişimi 1990'lara kadar uzanır ve başlangıçta BioVision'ın hareket yakalama sistemleri için özel bir dosya formatı olarak geliştirilmiştir. BVH'den önce hareket verileri kalem ve kağıtla kaydediliyor ya da bilgisayara manuel olarak giriliyordu. BVH, hareket verilerinin otomatik olarak yakalanmasını ve kaydedilmesini sağlayarak

gerçekçi animasyonlar oluşturmayı daha kolay ve verimli hale getirdi.

Günümüzde BVH, hareket yakalama verileri için yaygın olarak kullanılan bir dosya formatı olmaya devam etmektedir ve Autodesk'in Maya ve Blender gibi birçok animasyon yazılım paketi tarafından desteklenmektedir. Animasyon endüstrisinde, özellikle filmler ve video oyunları için gerçekçi ve gerçeğe yakın animasyonlar oluşturmak için önemli bir araç haline gelmiştir.[12]



Şekil 10: BVH iskelet yapısı
[13]

3.5 BVH Formatı Terminoloji

Hareketi tanımlamak için kullanılan anahtar kelimelerden bazıları aşağıda özetlenmektedir.

- **İskelet:** Hareketin temsil ettiği karakterin tamamı.
- **Kemik:** Hareket içindeki en küçük parça. İskeletler bir dizi kemikten oluşur.
- **Kanal veya özgürlük derecesi (DOF):** Bir kemiğin animasyon sırasında değiştirebileceği parametredir. Bir kemik, konum, yön ve ölçek olmak üzere üç ana eksenle hareket edebilir. Her eksen, bir DOF olarak kabul edilir..
- **Çerçeve:** Çerçeve, bir animasyondaki belirli bir zaman noktasını temsil eder. Her çerçeve, animasyondaki her bir kemiğin o zamandaki konumunu, yönünü ve ölçeğini tanımlar..

3.6 BVH Formatı Çalışma Prensipleri

Dosyanın hiyerarşik bölümü **HİYERARŞİ** anahtar kelimesi ile başlar ve bunu bir sonraki satırda **ROOT** anahtar kelimesi ve iskelet hiyerarşisinin kökü olan kemiğin adı takip eder. **ROOT** anahtar sözcüğü, yeni bir iskelet hiyerarşik yapısının başlangıcını belirtir ve BVH dosyası birçok iskelet içermeye kapasitesine sahip olmasına rağmen, genellikle dosya başına yalnızca tek bir iskeletin tanımlanması gerekir. İskeletin geri kalan yapısı yinelemeli bir yapıda tanımlanır. Her bir düğümün, **OFFSET**, **CHANNELS** ve **JOINT** gibi özellikleri vardır. **OFFSET**, Her eklem, ebeveyn eklemeye göre yer değiştirmesini belirtir.. **CHANNELS**, düğümün dönme bilgilerini temsil eder. **JOINT**, düğümün bir eklem noktasını temsil ettiğini belirtir ve altında başka düğümler bulunabilir. **Endvsite** ise Bir eklem sonunu belirtir.[14]

3.7 Eksik Verilerin Tamamlanması

Mediapipe ve Blender, karakter iskeleti tespiti ve animasyon oluşturma konularında farklı araçlar olsa da, bu iki platform arasında uyumluluk sağlamak oldukça önemlidir. Mediapipe kütüphanesi, insan vücudunun iskeletini tespit ederken bazı kemikleri eksik bırakmaktadır. Bu nedenle, eksik kemikleri tamamlayarak bunların x, y, z koordinatlarını bulmamız gerekmektedir. Eksik olan boyun kemiğini tespit etmek için sol dudak ve sağ dudak orta noktasını tespit edip ardından sağ omuz ve sol omuzun orta noktası tespit edilir. Ardından tespit edilen bu iki noktanın da orta noktası tespit edilip bu noktaya boyun kemiği olarak işaretlenir.

$$\begin{aligned} \text{Orta noktanın x koordinatı: } & \frac{x_1 + x_2}{2} \\ \text{Orta noktanın y koordinatı: } & \frac{y_1 + y_2}{2} \\ \text{Orta noktanın z koordinatı: } & \frac{z_1 + z_2}{2} \end{aligned}$$

```

mouth_right = results.pose_landmarks.landmark[mp_holistic.PoseLandmark.MOUTH_RIGHT.value]
mouth_left = results.pose_landmarks.landmark[mp_holistic.PoseLandmark.MOUTH_LEFT.value] #

right_shoulder = results.pose_landmarks.landmark[mp_holistic.PoseLandmark.RIGHT_SHOULDER.value]
left_shoulder = results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_SHOULDER.value]

# Mouth right ve left orta noktalarını hesapla
mouth_mid_x = (mouth_right.x + mouth_left.x) / 2
mouth_mid_y = (mouth_right.y + mouth_left.y) / 2
mouth_mid_z = (mouth_right.z + mouth_left.z) / 2

# Right shoulder ve left shoulder orta noktalarını hesapla
shoulder_mid_x = (right_shoulder.x + left_shoulder.x) / 2
shoulder_mid_y = (right_shoulder.y + left_shoulder.y) / 2
shoulder_mid_z = (right_shoulder.z + left_shoulder.z) / 2

# Neck noktasını hesapla
neck_x = (mouth_mid_x + shoulder_mid_x) / 2
neck_y = (mouth_mid_y + shoulder_mid_y) / 2
neck_z = (mouth_mid_z + shoulder_mid_z) / 2

print(f"neck: X={neck_x}, Y={neck_y}, Z={neck_z}")
cv2.circle(image, (int(neck_x * width), int(neck_y * height)), 5, (0, 255, 0), -1)

```

Şekil 11: Python da boyun kemiğinin konumunun hesaplanması.

Mediapipe kütüphanesi omurga kemiklerini tespit etmemektedir. Omurga kemiklerini hesaplamak için sol omuz ve sağ omuzun kemiklerinin orta noktası alındıktan sonra sol kalça ve sağ kalçanın orta noktası tespit edilir. Ardından tespit edilen bu iki nokta arasındaki mesafe 3 eşit noktaya ayırma işlemi yapılır. Tespit edilen noktalar opencv kütüphanesi ile işaretlenmiştir.

Şekil 12: Python da omurga kemiklerinin konumunun hesaplanması.

```

shoulder_mid_x = (right_shoulder.x + left_shoulder.x) / 2
shoulder_mid_y = (right_shoulder.y + left_shoulder.y) / 2
shoulder_mid_z = (right_shoulder.z + left_shoulder.z) / 2

# Kalçaların orta noktalarını hesaplayalım
hip_mid_x = (right_hip.x + left_hip.x) / 2
hip_mid_y = (right_hip.y + left_hip.y) / 2
hip_mid_z = (right_hip.z + left_hip.z) / 2

distance_x = abs(shoulder_mid_x - hip_mid_x)
distance_y = abs(shoulder_mid_y - hip_mid_y)

# Orta noktalar arasında 3 eşit aralık oluşturalım
point1_x = shoulder_mid_x + distance_x / 3
point1_y = shoulder_mid_y + distance_y / 3

point2_x = shoulder_mid_x + 2 * distance_x / 3
point2_y = shoulder_mid_y + 2 * distance_y / 3

# Yeni noktaları işaretleyelim
cv2.circle(image, (int(point1_x * width), int(point1_y * height)), 5, (0, 255, 0), -1)
cv2.circle(image, (int(point2_x * width), int(point2_y * height)), 5, (0, 255, 0), -1)

# Orta noktayı işaretleyelim
cv2.circle(image, (int(hip_mid_x * width), int(hip_mid_y * height)), 5, (0, 255, 0), -1)
cv2.circle(image, (int(shoulder_mid_x * width), int(shoulder_mid_y * height)), 5, (0, 255, 0), -1)

```

4 Sonuçlar

Hareket yakalama teknolojisi son yıllarda biyomekanik, yaya navigasyonu, eğitim ve simülasyon, sanal gerçeklik ve karakter animasyonu alanlarında oldukça önem kazanmıştır. Bu çalışmada da kameralardan alınan görüntü üzerinden vücut hareketlerini tespit edip bu hareketler 3 boyutlu olarak aktarılmıştır.

Kaynaça

- [1] İ. H. ÖZKİRİŞÇİ, “Motion capture teknolojisinin hareketli afişlerde kullanımına örnek bir çalışma,” *Fırat Üniversitesi Sosyal Bilimler Dergisi*, vol. 32, no. 3, pp. 1027–1041, 2022.
- [2] Ö. Ü. S. ERDEM, “Sanali gerçeğe dönüştürmede hareket yakalama teknolojisi,” 2021.
- [3] R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff, “Estimating 3d body pose using uncalibrated cameras, 2001, pp.”
- [4] Y. Desmarais, D. Mottet, P. Slangen, and P. Montesinos, “A review of 3d human pose estimation algorithms for markerless motion capture,” *Computer Vision and Image Understanding*, vol. 212, p. 103275, 2021.
- [5] Github, “mediapipe-pose-holistic.” <https://github.com/eminberkayd/mediapipe-pose-holistic>.
- [6] OpenAI, “Chatgpt.” <https://chat.openai.com/>.
- [7] G. Developers, “Pose landmark detection guide.” <https://developers.google.com/mediapipe/solutions/vision/pose>.
- [8] MOBİDİCTUM, “Mixamo nedir? mixamo nasıl kullanılır?” 2015.
- [9] Wikipedia, “Blender.” https://tr.wikipedia.org/wiki/Blender_yaz
- [10] <https://www.mixamo.com>.

- [11] ROKOKO, “Richer animations start with rokoko mocap.” <https://www.rokoko.com/>.
- [12] Wikipedia, “Biovision hierarchyd,” 2023. https://en.wikipedia.org/wiki/Biovision_Hierarchy.
- [13] M. Technologies and . Applications (Semester A, “Bvh motion capture data animated,” 2007. <https://www.cs.cityu.edu.hk/howard/Teaching/CS4185-5185-2007-SemA/Group12/BVH.html> [Accessed: (Use the date of access)].
- [14] M. Meredith and S. Maddock, “Motion capture file formats explained,” *Production*, 01 2001.