

T.C KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



Videodan Animasyon Oluşturulması

İlknur Koparır

Danışman

Dr.Öğr.Üyesi Emre Güngör

**Yapay Zeka & Veri Odaklı Sistem Tasarımı & Latex İle Rapor
Hazırlama**

13.06.2024

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. 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 ayırt eden algoritmalar 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.



(a)



(b)

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 Blacton'un, Şekil 2'de görüldüğü üzere ilk çizgi filmi olan Humorous Phases 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 yakalamanın 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 kullanılarak 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 görünümüler arasında 2B eklem konumları sağlar.

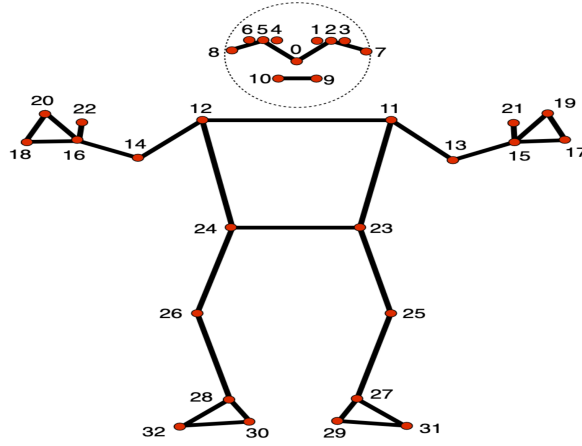
Kakadiaris ve Metaxas [4], dik açılı konfigürasyonlarda çoklu kamera görüntülerini kullanarak insan vücut parçalarının 3B model tabanlı izleme ve şekil tahmini için bir yöntem sunarlar. Daha sonra yaklaşımı, çoklu görüşlü video dizilerinden insan hareketinin tahmini ve buna göre animasyon dizileri oluşturmak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

3 Yöntem

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’ın 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.



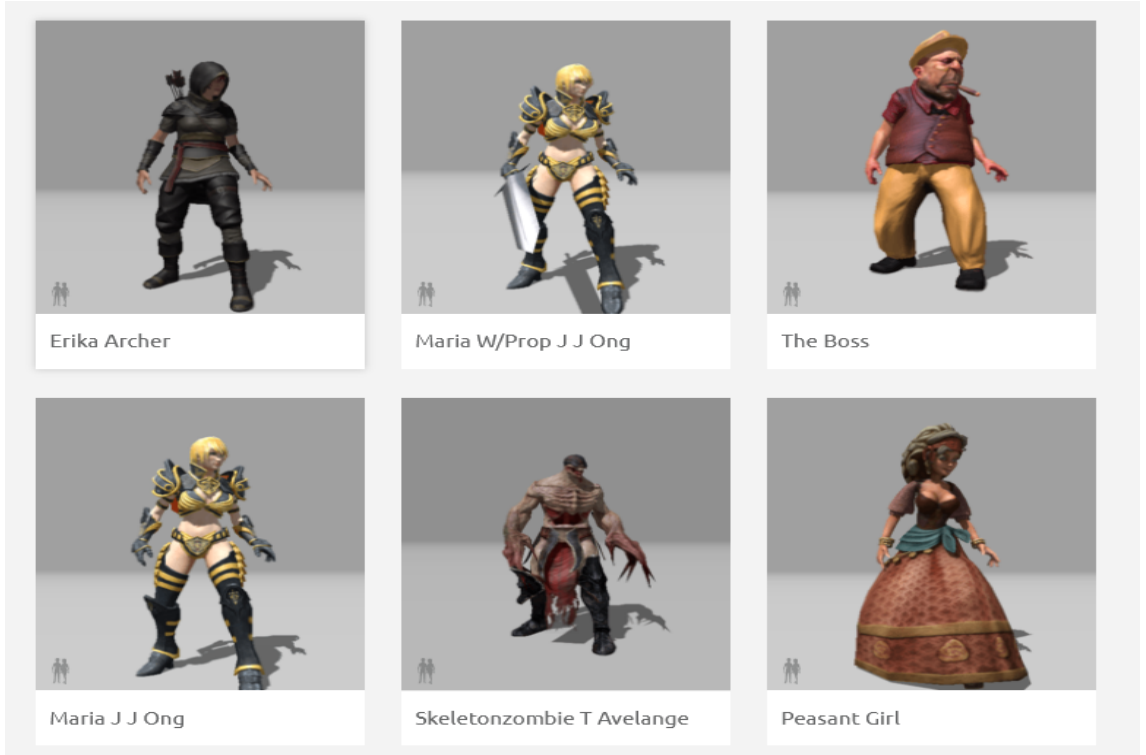
Şekil 3: MediaPipe’da 33 vücut parçasının konumu [5]

3.2 Blender ve Mixamo

Mixamo, Adobe tarafından sunulan bir hizmet olup, sanat projelerinde, filmlerde ve oyunlarda kullanılmak üzere herkesin erişebileceği çevrim içi bir karakter ve mocap animasyon veritabanıdır.[7] 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.[8]

3.3 3D Karakter Modeli

Animasyon projeleri için özel karakter modeli bulunmuyorsa Mixamo kullanmamız için önceden hazırlanmış karakterler sunmaktadır. Karakterler projede kullanılmak üzere .fbx formatında indirilmesi gerekmektedir.



Şekil 4: Mixamo da bulunan hazır karakterlerden bazıları
[9]

3.4 Blender ortamında karaktere iskelet yerleştirilmesi

Mixamo'dan indirilen karaktere hareket vermek için Blender üzerinden iskelet yerleştirme işlemi yapılmıştır. Bu işlem, karakterin hareket edebilmesi ve animasyonlarının uygulanabilmesi için gerekli temel yapıyı oluşturur. İskelet, karakterin vücut yapısına uygun olarak düzenlenerek eklem bölgeleri doğru şekilde konumlandırılmıştır.



Şekil 5: Blender ortamında karaktere iskeletin yerleştirilmesi ve ağırlıklandırılmasının yapılması.

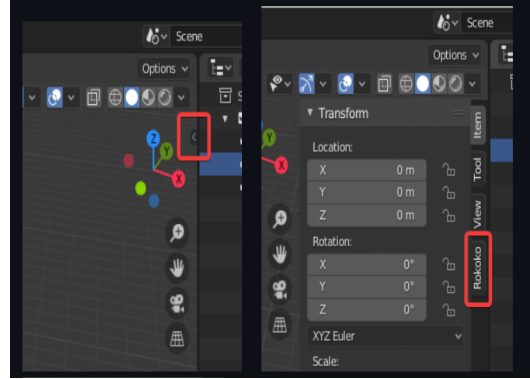
3.5 Rokoko

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

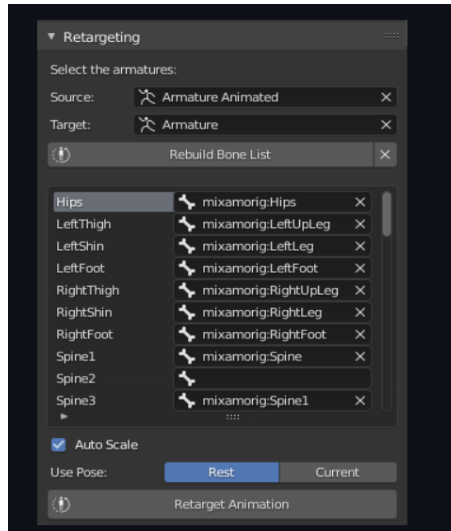
3.5.1 Blender Ortamında Karaktere Hareket Verilmesi

Karaktere hareket verilmek üzere blender da Retargeting paneli açılmıştır. Bu panel, animasyonların yeniden hedeflenmesi işlemlerini gerçekleştirmemize olanak sağlamaktadır. Ardından animasyonun kaynak iskeletini ve animasyonun hedef alınacağı iskelet seçilmiştir. Kaynak iskelet, animasyonun geldiği

orjinal 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" işlemi yapılmıştır. Kemiklerin doğru eşleştirildiğinden emin olmak için Blender'ın oluşturduğu kemik listesini kontrol edilmiştir. Eğer eksik veya yanlış kemikler varsa, bunların düzeltilmesi yapılır. Ardında animasyona hareket verilmiştir.



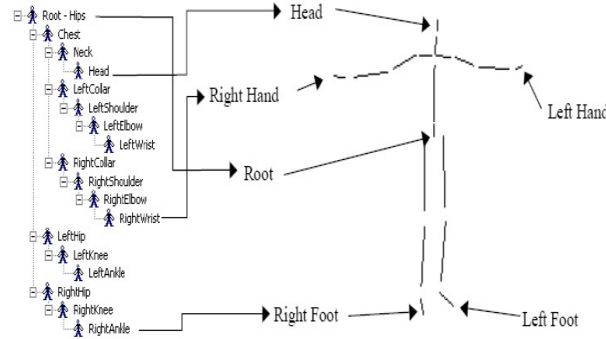
Şekil 6: Retargeting.



Şekil 7: İki iskelet arasında kemiklerin eşleştirilmesi.

3.6 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



Şekil 9: BVH iskelet yapısı
[11]

- **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 eksende 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..

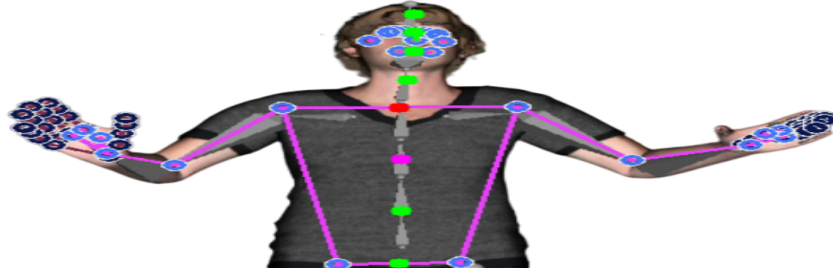
3.8 BVH Formatı Çalışma Prensibi

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.[12]

4 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.

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



Şekil 10: Eksik verilerin tamamlanıp kordinatlarının işaretlenmesi

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.

4.1 ANTROPOMETRİK ÖLÇÜMLER

Antropometri, Yunanca antropos (insan) ve metrein (belirleme, ölçü) sözcüklerinden oluşan ve insan vücudunun ölçülerini konu edinen bir bilim dalıdır. Bireyler veya gruplar arasında, anatomi, coğrafi bölge ve meslek grupları gibi çeşitli

```

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.

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)

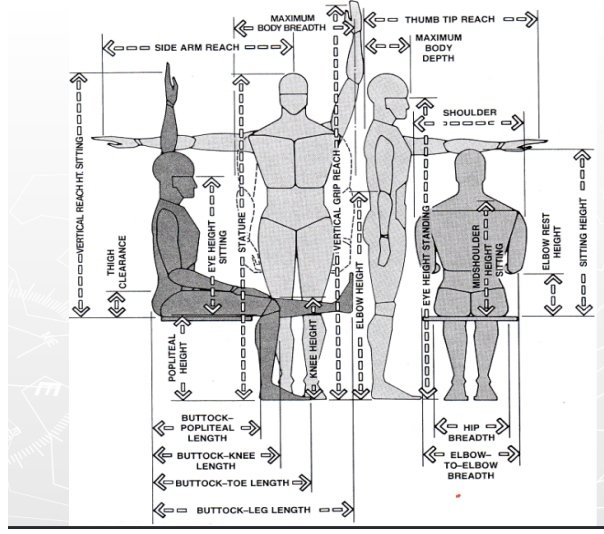
# Orta noktayı işaretleyelim
cv2.circle(image, (int(hip_mid_x * width), int(hip_mid_y * height)), 5, (0, 255, 0), -1)
cv2.circle(image, (int(shoulder_mid_x * width), int(shoulder_mid_y * height)), 5, (0, 255, 0), -1)

```

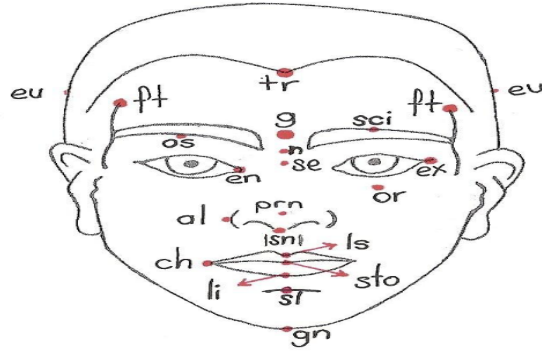
faktörlerden kaynaklanan farklılıkları ve benzerlikleri saptayarak daha geniş bir insan kitlesine uygun tasarımlar yapma imkânı sağlar.

4.1.1 Baş Bölgesindeki Antropometrik Noktalar

Başın konumu Oksipital çıkıntı ve kaşların üzerinden ölçülür.



Şekil 13: Ölçüm noktaları.

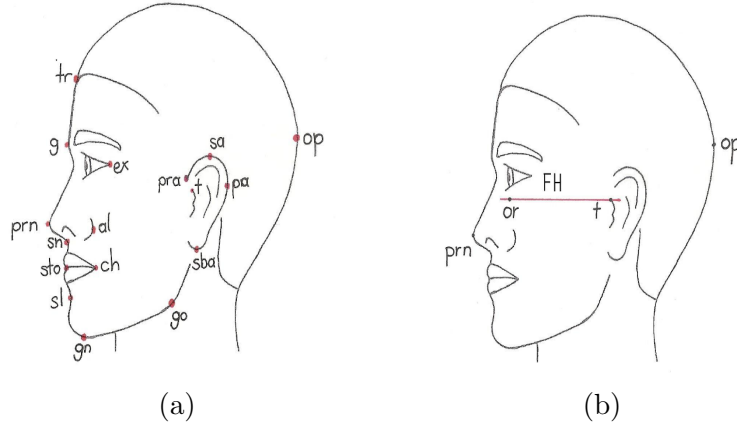


Şekil 14: Önden görünüşte baş üzerindeki antropometrik noktalar.

4.2 Altın Oran

Temelde, bir nesnenin veya bir yapının estetik olarak hoş ve dengeli görünmesini sağlamak için kullanılan bir oranı ifade eder. Eski Mısırlılar ve Yunanlılar tarafından keşfedilmiş, mimaride ve sanatta kullanılmıştır. Sanatçılar, bilim adamları ve tasarımcılar, araştırmalarını yaparken ya da ürünlerini ortaya koyarlarken orantıları altın orana göre belirlenmiş insan bedenini ölçü olarak alırlar. Altın oran Deniz kabuğunun spiralinde, insan elinde, DNA diziliminde, Akciğerlerde ve uzayda olmak üzere pek çok yerde karşımıza çıkmaktadır.

$$\phi = \frac{1+\sqrt{5}}{2}$$



Şekil 15: (a) Yandan görünüşte, baş üzerindeki antropometrik noktalar. (b) Başın standart pozisyonu FH

4.2.1 Yüzdeki Altın Oran

Yüzde estetik bir görünüm elde etmek amacıyla kullanılan matematiksel bir orandır. Bu oran, yüzdeki farklı bölgeler arasındaki oranları belirleyerek yüz hatlarının dengesini ve simetrisini değerlendirir. Altın Oran, genellikle gözler, burun ve ağız gibi temel yüz özellikleri arasındaki ilişkiyi inceler.

- Alın yüksekliği Yüz uzunluğunun yüzde 33'üne eşit olmalıdır.
- Burun uzunluğ Alın yüksekliğinin yüzde 50'sine veya burun kökü ile çene ucu arasındaki mesafenin 33'üne eşit olmalıdır.

5 Rotasyon İşlemleri

Mediapipe ve Blender gibi farklı platformlar, farklı koordinat sistemlerini kullanır. Bu sistemlerdeki koordinatlar farklı yönlere veya orijinlere sahiptir. Mediapipe'de genellikle kamera görüş açısına göre tanımlanan bir koordinat sistemi kullanılırken, Blender'da dünya uzayına göre bir koordinat sistemi kullanır. Bu nedenle, Mediapipe'den gelen koordinatlar doğrudan Blender koordinatlarına uygulanamaz. Mediapipe'den gelen koordinatları Blender koordinat sistemine dönüştürmek için dönüşüm matrisleri kullanıyoruz.

Dönüşüm Matrisi Uygulanmadan Önce

x: 0.52258945
y: 0.2965024
z: -0.23053758

Dönüşüm Matrisi Uygulandıktan Sonra

x: 0.52258945

y: -0.23053758

z: 0.2965024

Dönme matrisin amacı Öklid uzayında vektörlerin rotasyonunu gerçekleştirmektir. Genellikle uzayı belirli bir açı veya derece etrafında döndürmek için kullanılır. İki veya üç boyutlu uzayda nesnelerin konumlarını değiştirmek, yönelimlerini değiştirmek ya da vektörün Kartezyen koordinatlarını yeni korinatlarla eşlemek istediğimizde dönme matrisleri kullanılır. Dönme matrisleri, bilgisayar grafikleri, robotik, bilgisayar görüşü gibi alanlarda yaygın olarak kullanılır. Özellikle 3D grafiklerde nesnelerin dönme ve yönlendirme işlemlerinde sıkça karşımıza çıkarlar.

2D dönme matrisi:

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

3D dönme matrisi:

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

6 Euler Formülü

Euler dönüşleri, bir nesnenin 3D uzayda dönüşünü ifade etmek için kullanılan bir yöntemdir. Bu yöntemde, bir nesnenin dönme hareketi, üç farklı eksen etrafında gerçekleşen açısal değişikliklerle tanımlanır. Euler dönüşleri, nesnenin dönüşünü ifade etmek için üç açı kullanır bunlar sırasıyla roll, pitch ve yaw açılarıdır.

- Roll açısı x eksenindeki dönüşü temsil eder.
- Pitch açısı y eksenindeki dönüşü temsil eder.
- Yaw açısı z eksenindeki dönüşü temsil eder.

7 Gimbal Lock

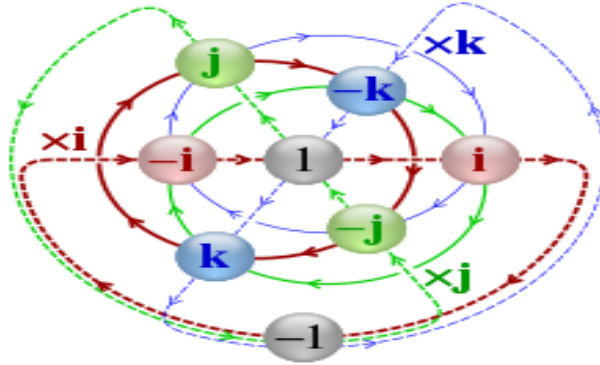
Gimbal lock (Gimbal kilidi), 3D uzayda nesnelerin dönüşlerini ifade etmek için kullanılan Euler açılarının bazı durumlarda ortaya çıkan istenmeyen bir durumdur.

Gimbal sistemi, bir nesnenin üç eksen etrafında dönmesine izin veren bir mekanizmadır. Her eksen bir gimbal olarak adlandırılır ve bu eksenler birbirine dik olarak yerleştirilir. Ancak, Euler açıları kullanılarak nesnenin rotasyonu ifade edilirken, bu gimballer bazen birbirine hizalanabilir. Bu durumda, iki veya daha fazla eksen aynı hizada olduğunda, sistemde gimbal lock oluşur. Bu durumda, belirli bir dönüş yönü veya açısal hareket kaybolabilir veya kısıtlanabilir. Özellikle, iki gimbal birbirine hizalandığında, üçüncü bir eksen etrafındaki dönüşün doğru bir şekilde ifade edilemez.

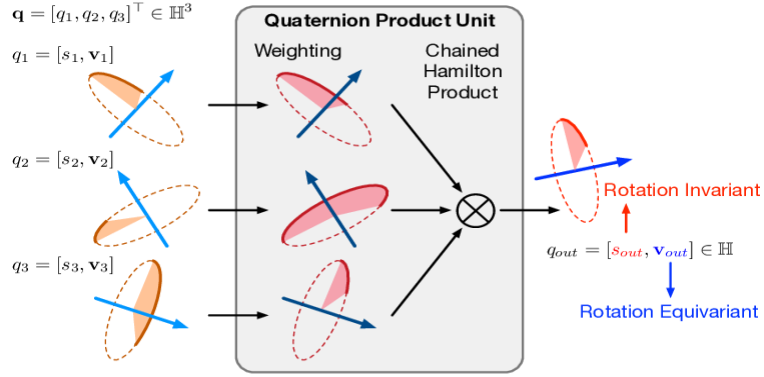
8 Quaternion

Euler açıları objelerin rotasyonlarını çözme konusunda insanın aklına gelen en doğal yöntem olsa da en optimal yöntem değil. Örneğin “Gimbal Lock” denilen problem euler açılarını kullandığımızda ortaya çıkıyor. Eksenlerimizdeki rotasyonlar üst üste bindiğinde sistemimiz kilitleniyor ve rotasyonları gösterme gücümüz azalıyor.

Bu sorunun yaşanmaması için Quaternion’lar kullanılıyor. Quaternion’larda Euler açılara ek olarak 4. bir değer var. Bu değer skaler bir değer. Ve bu 4. değer sayesinde çakışmalar önleniyor. Kuvaterniyonlar (quaternion’lar), üç boyutlu uzayda dönüşümleri temsil etmek için kullanılan matematiksel yapılar olup kompleks sayılar üzerine genişletilmiş bir sayılar sistemidir. William Rowan Hamilton tarafından 1843 yılında geliştirilmiştir. Kuvaterniyonlar, vektörlerin döndürülmesi, uzay geometrisinde rotasyonların tanımlanması ve fiziksel simülasyonlarda kullanılırlar

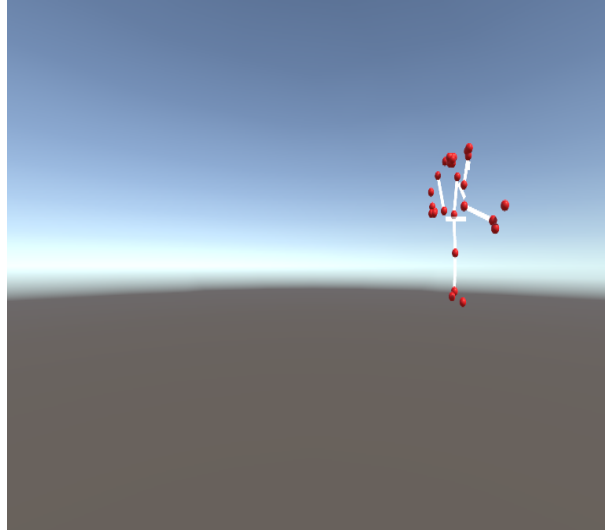


$$q = a + bi + cj + dk$$



9 Unity ortamında iskelete hareket verilmesi

Vücut noktalarının x, y, z koordinatlarını kaydetmek ve bu verileri Unity ortamında kullanmak amaçlanmıştır. İlk olarak, 33 adet vücut noktasının koordinatlarını içeren bir animation.txt dosyası oluşturduk. Daha sonra, Unity ortamında her biri bir eklemi temsil eden 33 adet küre ekledik. Kaydettiğimiz animation.txt dosyasını, projedeki Asset klasörüne yapıştırarak Unity’de kullanılabilir hale getirdik. animation.txt dosyasındaki string formatındaki değerleri koordinatlara dönüştürmek amacıyla bir C projesi oluşturduk. Her yerden erişilebilir bir Vücut adında game object oluşturduk ve bu game object’i yönetici bir script içine yerleştirdik. 33 adet landmark olduğu için Vücut nesnesinin değerini 33 yaparak, oluşturduğumuz kürelerle bu noktaları eşleştirdik.



Şekil 16: Unity ortamında motion capture.

10 Alınan Hatalar

Euler açısı kullanarak eklemelerin rotasyon hesaplaması yaparken çıkan sonuçlar hep 0 ve 1 arasında değerlerdir. Ancak hazır bvh dosyaları incelendiğinde eklemelerin rotasyon değerleri 0 ve 1 arasına sıkışmamış, çeşitli değerler almış.

```
Frames: 250
Frame Time: 0.041667
2.828610 -2.035149 2.642982 -1.282020 3.466291 -6.774150 0.000001 0.055651 0.221809 -4.972735 -0.
0.020964 0.158310 18.467851 1.337458 10.657977 -19.574564 72.162417 -24.672421 -0.000199 0.000139
0.000001 0.000002 -0.000004 -0.000001 -0.000002 0.000000 0.000000 0.239058 0.002702 0.004759 -0.0
0.047726 0.004758 -0.000004 -0.000001 0.000001 -0.000004 0.000001 -0.000001 -0.000001 -0.000000 -
0.000001 -0.000000 -0.000003 -0.000004 0.000002 0.170583 -0.108633 -0.080056 -0.000012 -0.000027
11.384879 -5.597772 -4.079387 -0.144213 0.020963 0.158307 12.640689 -3.237309 -14.785579 -19.0440
6.189920 -0.235456 -0.043552 0.004756 0.000001 0.000000 -0.000001 -0.000002 -0.000001 -0.000001 0
0.000000 -0.000000 -0.000001 -0.000002 0.000000 0.000002 -0.223809 0.047725 0.004755 0.000001 -0.
0.004757 0.000001 -0.000001 0.000000 0.000000 0.000000 -0.000003 0.000001 -0.000004 0.000002 -0.1
0.206316 0.040529 -0.157451 -11.313028 -5.688955 -5.736538 28.394684 -0.000006 -0.000009 -19.3943
0.040529 -0.157452 -1.687041 -5.166931 -3.267179 12.050102 -0.000004 -0.000012 -21.368230 20.5256
2.826553 -2.034858 2.643260 -1.345260 3.419777 -6.821706 0.000001 0.055651 0.221809 -4.974376 -0.
0.158311 18.408673 1.363250 10.582374 -19.584591 72.066576 -24.817846 -0.000199 0.000139 -75.0105
0.000002 0.000001 -0.000001 -0.000002 -0.000003 -0.000002 0.239059 0.002701 0.004758 -0.000001 -0
0.047725 0.004757 0.000000 0.000001 0.000000 -0.000001 -0.000000 0.000000 0.000000 0.000001 0.000
0.000001 -0.000001 -0.000000 -0.000002 -0.000002 -0.000005 0.000000 0.000001 0.170583 -0.108632 -
2.074620 11.424907 -5.559226 -4.070977 -0.144213 0.020962 0.158308 12.605654 -3.135293 -14.717473
5.971013 -0.235456 -0.043552 0.004756 0.000000 -0.000000 -0.000002 -0.000000 -0.000001 -0.000001
0.000001 0.000001 -0.000002 0.000003 -0.000000 0.000001 -0.223809 0.047725 0.004755 0.000002 -0.0
0.004757 -0.000001 -0.000000 -0.000002 0.000003 0.000001 -0.000001 -0.000000 -0.000006 0.000001 -
0.206316 0.040529 -0.157452 -11.398746 -5.628402 -5.450453 28.802587 -0.000006 -0.000008 -19.4036
```

Şekil 17: Olması gereken bvh formatı.

```
Frames: 250
Frame Time: 0.041667
0.397457 -0.000038 0.476630 0.000000 0.000000 0.000000 0.414667 -0.205781 0.300554 0.812124 0.77718
0.296502 0.948730 0.714440 0.187400 1.108974 0.681946 -0.315817 1.297112 0.820325 -0.375872 1.42106
0.306745 -0.181024 0.304605 0.732885 0.722543 0.236914 0.562932 0.698561 -0.220895 0.399093 0.84277
0.954174 -0.028790 0.906018 1.101729 -0.090687 0.913277 1.373944 0.141803 0.902091 1.558697 0.09875
0.755691 1.361587 0.275586 0.784567 1.538563 0.273486 0.397904 -0.000031 0.478517 0.000000 0.000000
0.168644 0.845844 0.779829 0.779829 0.522898 -0.208077 0.297035 0.948600 0.716483 0.211372 1.109271
0.700203 0.051526 0.805203 0.484051 -0.984685 0.306267 -0.175396 0.305589 0.731969 0.725037 0.24405
0.455251 -0.024592 0.479520 0.853155 0.958037 -0.024624 0.907655 1.104215 -0.080521 0.915359 1.3744
0.024498 0.697178 1.098608 0.028279 0.758501 1.362190 0.284179 0.790076 1.540623 0.314913 0.398216
0.860134 -0.162677 0.839960 0.741191 0.237631 0.852396 0.860134 0.860134 0.523986 -0.185462 0.33702
1.310492 0.932826 -0.483242 0.840843 0.773842 0.119317 0.816343 0.547266 -0.922574 0.309755 -0.1399
0.903818 -0.423435 0.283345 0.957022 -0.359064 0.455864 -0.018291 0.520429 0.854080 1.039704 -0.018
0.340568 0.018337 0.518121 0.738784 1.037395 0.018361 0.701098 1.172315 0.043297 0.762423 1.425157
0.401879 -0.153286 0.367524 0.803390 0.913388 -0.153276 0.803882 0.794494 0.273685 0.804128 0.91338
0.807397 -0.348905 1.182418 0.947731 -0.622274 1.254035 1.007120 -0.464535 0.803088 0.826322 0.1658
0.522960 0.828776 -0.139750 0.365764 0.967726 -0.290378 0.259223 1.026109 -0.266033 0.460529 -0.035
0.235339 0.920651 1.640347 0.301436 0.342492 0.035483 0.547032 0.744002 1.092897 0.035492 0.691561
0.542769 0.000000 0.000000 0.000000 0.398101 -0.242972 0.364616 0.794589 0.907384 -0.242957 0.79674
0.843931 0.208241 1.059993 0.802143 -0.460957 1.184594 0.945462 -0.643435 1.269453 1.004973 -0.4576
0.688322 0.852068 0.272582 0.519507 0.824859 -0.306530 0.361678 0.964409 -0.426922 0.248770 1.02289
1.218265 -0.086931 0.923306 1.474587 0.239108 0.917506 1.645923 0.304870 0.338002 0.034961 0.543457
0.761015 1.624528 0.455827 0.392620 -0.000013 0.539336 0.000000 0.000000 0.000000 0.394471 -0.29984
0.902751 0.502071 -0.325120 0.360004 0.897776 0.840700 0.155575 1.055235 0.800503 -0.527040 1.19735
0.590525 -1.253722 0.286870 -0.274569 0.366827 0.682575 0.847523 0.206126 0.515543 0.821744 -0.4220
```

Şekil 18: Euler açısı kullanarak elde ettiğim bvh dosyası.

11 Bulgu ve Tartışma

Vize öncesi hazırlanan gantt şemasında bvh dosya formatına 14 gün ayrılmıştır. Ancak projenin ilerleyen kısımlarında bizden eklemlerin x, y, z kordinatlarını değil x,y,z rotasyonlarını istediğini fark ettim. Bu yüzden kemiklerin euler açısı ile dönme kordinatlarını hesaplamaya çalıştım. Bu hesaplamaları yaparken birçok sorun ile karşılaştım ve dönme noktaları düzgünce hesaplanamadı. Bundan dolayı proje süresi uzamıştır.

12 Beklenen Sonuçlar

Proje henüz tamamlanmadığı için beklenen sonuçlar yazılmıştır. Hareket yakalama teknolojisi son yıllarda biyomekanik, yaya navigasyonu, eğitim ve simülasyon, sanal gerçeklik ve karakter animasyonu alanlarında oldukça önem kazanmıştır. Bu çalışmada da hedeflenen sonuç kameradan alınan görüntü üzerinden vücut hareketlerini tespit edip bu hareketler 3 boyutlu olarak aktarmaktır.

Kaynakça

- [1] İ. H. ÖZKİRİŞÇİ, “Motion capture teknolojisinin hareketli afişlerde kullanıma 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] G. Developers, “Mediapipe.” <https://ai.google.dev/edge/mediapipe>, Year Published/ 2024-05-14. Accessed: 06.03.2024.
- [6] OpenAI, “Chatgpt.” <https://chat.openai.com/>, Year Published/ 2015–2024.
- [7] MOBİDİCTUM, “Mixamo nedir ?.” <https://mobidictum.com/tr/mixamo-nedir/>, Year Published/ 2015–2024.

- [8] Wikipedia, “Blender(yazılım).” [https://tr.wikipedia.org/wiki/Blender_\(yaz%C4%B1l%C4%B1m\)](https://tr.wikipedia.org/wiki/Blender_(yaz%C4%B1l%C4%B1m)), Year Published/ 15 Mayıs 2024.
- [9] Adobe, “Get animated.animate 3d characters for games, film, and more..” <https://www.mixamo.com/>), Year Published/ 2024.
- [10] Wikipedia, “Biovision hierarchy.” https://en.wikipedia.org/wiki/Biovision_Hierarchy), Year Published/ 2003.
- [11] M. Technologies and . Applications (Semester A, “Bvh motion capture data animated.” <https://www.cs.cityu.edu.hk/~howard/Teaching/CS4185-5185-2007-SemA/Group12/BVH.html>, Year Published/ 2007.
- [12] M. Meredith and S. Maddock, “Motion capture file formats explained,” *Production*, 01 2001.