04-landmarks_position.ipynb

Pablo Munarriz Senosiain

5 de diciembre de 2024

1. Introducción

El propósito de este documento es explicar el código del cuaderno '04-landmarks_position.ipynb'. En este cuaderno de Jupyter se definen varias funciones con las que se pretende establecer relaciones entre los 'pose landmarks' de MediaPipe y varios atributos naturales de los brazos. De esta forma, trataríamos de generar el movimiento de los brazos entre dos posturas.

2. Pose Landmarks

MediaPipe ofrece, entre otras cosas, modelos de detección de puntos de referencia sobre personas. Concrétamente, el modelo 'pose lanmarker model' detecta 33 puntos sobre los cuerpos humanos, y su objetivo es detectar posturas.

Al aplicar el modelo a una imagen, obtenemos un objeto del tipo 'mediapipe.framework.formats.landmark_pb2.NormalizedLandmarkList'. Este objeto, tal y como dice el nombre, es una lista de 'landmarks normalizados'. Cada uno de estos landmarks contiene hasta cuatro atributos: 'x', 'y', 'z' y 'visibility'. Aquí podemos ver los detalles de cada atributo:

x and y: Landmark coordinates normalized between 0.0 and 1.0 by the image width (x) and height (y).

z: The landmark depth, with the depth at the midpoint of the hips as the origin. The smaller the value, the closer the landmark is to the camera. The magnitude of z uses roughly the same scale as x.

visibility: The likelihood of the landmark being visible within the image.

En nuestro caso, almacenamos los valores de estos atributos en un array de NumPy, por lo que los objetos 'NormalizedLandmarkList' acaban siendo arrays bidimensionales, donde la primera dimensión representa cada landmark y la segunda dimensión cada atributo. Concretamente, los 'pose_landmarks' se almacenan en arrays de dimensión $(33, 3)$ ó $(33, 4)$, en función de si almacenamos la visibilidad o no. El 'dtype' que usamos es 'np.float64'.

Por otra parte, para entender la posición de los brazos nos basta únicamente con los landmarks del torso y brazos, por lo que también solemos trabajar con objetos de nombre 'half_pose_landmarks', que básicamente son los subarrays de los ‘pose_landmarks’ que contienen la información de los landmarks que queremos. Concretamente, la relación es ‘half_pose_landmarks = pose_landmarks[11:25]’.

3. Atributos

Para entender la posición de los brazos, en primer lugar consideramos una base para cada brazo. El primer vector de la base es el vector unitario que tiene la misma dirección que el vector entre los hombros y que, al situarlo en el hombro en cuestión, apunta hacia fuera del cuerpo. El segundo vector de la base es también unitario, ortogonal al primero, contenido en el plano del torso y en sentido de los hombros a las caderas. El tercer vector es el necesario para que las bases sean ortonormales, por lo que es el producto vectorial del primer vector con el segundo. Así, el tercer vector de la base del brazo izquierdo sale del hombro ortogonal al plano de la espalda (hacia atrás), mientras que el tercer vector de la base del brazo derecho sale del hombro ortogonal al plano del torso (hacia adelante). En la Figura 1 vemos una representación gráfica de ambas bases.

También consideraremos una base para cada mano. En este caso, el primer vector de la base es el vector unitario normal a la palma de la mano (saliendo de la palma). El tercer vector será el que apunte de la muñeca hacia el centro de la palma. Y el segundo, para hacer una base ortonormal, será el producto vectorial del tercer vector con el primero. De nuevo, el sentido del segundo vector cambia en función del lado: para la mano izquierda, el segundo vector apunta al pulgar; para la mano izquierda, al meñique.

Los atributos que consideraremos para tratar de entender la posición de cada brazo serán:

- Dirección del hombro: vector unitario que apunta del hombro al codo expresado en la base del brazo.
- Longitud del brazo (superior): distancia entre el hombro y el codo.
- Rotación del hombro: en radianes, respecto a alguna posición concreta.
- Ángulo del codo: en radianes.
- Longitud del antebrazo: distancia entre el codo y la muñeca.
- Rotación del codo: en radianes, respecto a alguna posición concreta.
- Rotación e inclinación de la muñeca: coordenadas esféricas del vector unitario que representa al antebrazo en la base de la mano.
- Vectores del meñique, índice y pulgar en la base de la mano.

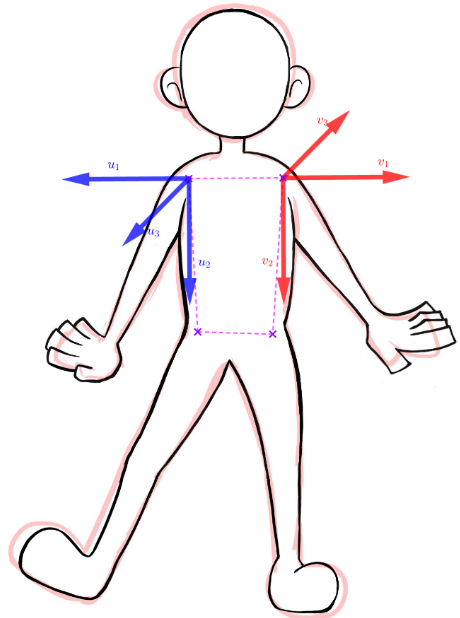

Figura 1: Representación gráfica de las bases de los brazos.

4. Preliminares

Antes de empezar con las funciones que traducen de landmarks a atributos y viceversa, vamos a comentar algunas funciones previas que conviene definir.

4.1. azimuth y colatitude

Estas funciones devuelven el ángulo acimutal y la colatitud de un vector real tridimensional y unitario, respectivamente. Tienen tres parámetros de entrada: ‘x’, ‘y’ y ‘z’, que representan las coordenadas del vector.

La función ‘azimuth’ devuelve un número en el intervalo $[-\pi, \pi)$ que mide el ángulo en radianes entre el semieje positivo X y el vector $(x, y, 0)$, siendo positivo cuando $y > 0$. Cabe destacar que cuando ‘xé ‘y’son cero, la función devuelve cero.

Por otra parte, colatitude’ devuelve un número en el intervalo $[0, \pi]$ que mide el ángulo en radianes entre el semieje positivo Z y el vector (x, y, z). En la Figura 2 tenemos el código de cada función.

4.2. max shoulder rotation

Las siguientes dos funciones se llaman ‘max_left_shoulder_rotation’ y ‘max_ right_shoulder_rotation’. El objetivo de estas funciones es, dada la dirección del

```python
def azimuth(x, y, z):
    """
    Given a unit 3D real vector, returns the azimuth of the vector.
    """
    if np.isclose(x, 0):
        if np.isclose(y, 0):
            return np.float64(0)
        else:
            return np.sign(y) * pi / 2.
    else:
        aux = np.arctan(y/x)
        if x>0:
            return aux
        elif y>0:
            return aux + pi
        else:
            return aux - pi

def colatitude(x, y, z):
    """
    Given a unit 3D real vector, returns the colatitude of the vector.
    """
    return np.arccos(np.clip(z, -1, 1))
```

Figura 2: Código de las funciones azimuth y colatitude.

hombro, devolver el vector normal al plano que contiene el brazo cuando el hombro está rotado al máximo (en sentido antihorario para el hombro izquierdo y horario para el hombro derecho).

Estas funciones se basan en cuatro pasos:

1. Construir un vector auxiliar que represente la dirección del antebrazo en una posición determinada.

2. Calcular el ángulo entre el vector auxiliar y la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo.

3. Calcular la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo.

4. Devolver el producto tensorial entre la dirección del hombro y la dirección del antebrazo del paso anterior.

Vayamos punto por punto, empezando por el primero. La idea para la elección del vector auxiliar es muy sencilla, simplemente tomaremos la dirección del

```python
if np.isclose(y, 0):
    aux_vector = np.array([0, -1, 0], dtype=np.float64)
elif np.isclose(y, 1):
    aux_vector = np.array([0, 0, -1], dtype=np.float64)
elif np.isclose(y, -1):
    aux_vector = np.array([0, 0, 1], dtype=np.float64)
else:
    aux_vector = np.sign(y) * np.array([x, y - 1./y, z], dtype=np.float64)
    aux_vector /= np.linalg.norm(aux_vector)
```

Figura 3: Código del vector auxiliar de la función 'max_left_shoulder_rotation'.

antebrazo que es ortogonal a la del hombro (es decir, fijamos el ángulo del codo a 90 grados) y tal que las proyecciones de las dos direcciones sobre el plano $y = 0$ tienen la misma dirección. Debemos considerar algunos casos particulares: en primer lugar, si la dirección del hombro ya está sobre el plano $y = 0$, entonces el vector auxiliar será $(0, -1, 0)$ (el antebrazo apunta hacia arriba, si estamos erguidos); en segundo lugar, si la dirección del hombro es $(0, 1, 0)$ (apunta hacia el suelo, si estamos erguidos), entonces su proyección sobre el plano $y = 0$ es el vector nulo, con lo que decidimos que el vector auxiliar sea el que apunta hacia el frente, por lo que será $(0, 0, -1)$ en el brazo izquierdo y $(0, 0, 1)$ en el derecho; por último, si la dirección del hombro es $(0, -1, 0)$, decidimos que el vector auxiliar sea el que apunta hacia atrás, por lo que será $(0, 0, 1)$ en el brazo izquierdo y $(0, 0, -1)$ en el derecho.

Supongamos ahora que la dirección del hombro es (x, y, z), con y distinto de 0, 1 y -1. Si $y > 0$, entonces queremos que el vector auxiliar sea proporcional a uno de la forma (x, y', z). Como además queremos que los dos vectores sean ortogonales, debe pasar que $x^2 + yy' + z^2 = 0$, por lo que

$$y' = -\frac{x^2 + z^2}{y} = \frac{y^2 - 1}{y} = y - 1/y\,.$$

En cambio, si $y < 0$, entonces lo que queremos es que el vector auxiliar sea proporcional a uno de la forma $(-x, y', -z)$. En este caso, $-x^2 + yy' - z^2 = 0$ implica

$$y' = \frac{x^2 + z^2}{y} = \frac{1 - y^2}{y} = 1/y - y\,.$$

En definitiva, el vector auxiliar es de la forma

$$\text{signo}\,(y)\,(x, y - 1/y, z)\ .$$

En la figura 3 vemos el código que calcula este vector auxiliar en la función 'max_left_shoulder_rotation'. Cabe decir que la variable 'pi'se ha definido previamente como 'np.float64(np.pi)'.

Vamos con el segundo paso. Tenemos que obtener el ángulo entre el vector auxiliar que acabamos de definir y la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo. En este caso vamos a ir brazo por brazo, empezando por el izquierdo. El ángulo lo consideraremos en sentido antihorario respecto a la dirección del hombro. La idea para obtener este ángulo, cualquiera que sea la dirección del hombro, consiste en estimar (a ojo) su valor para ciertas direcciones del hombro y asumir una especie de linealidad en distintos sectores.

En primer lugar, vamos a suponer que cuando la dirección del hombro es $(0, 1, 0)$, el ángulo que buscamos es $\pi/4$. Es decir, que cuando la parte superior del brazo apunta hacia el suelo (si estamos erguidos), la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo se separa del vector auxiliar (el que apunta hacia el frente) 45 grados.

En segundo lugar, vamos a suponer que, cuando la dirección del hombro es de la forma $(\cos\varphi, 0, \sin\varphi)$, con $\varphi \in [-\pi, 0]$, el ángulo es 0. Es decir, que cuando la dirección del hombro es paralela al suelo (si estamos erguidos) y no apunta hacia atrás, la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo es precisamente el vector auxiliar, por lo que apunta hacia arriba.

En tercer lugar, vamos a suponer que, cuando la dirección del hombro es $(0, 0, 1)$ (apunta totalmente hacia atrás), el ángulo es $-\pi/2$. Es decir, que la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo es $(1, 0, 0)$.

Por último, vamos a suponer que, cuando la dirección del hombro es $(0, -1, 0)$ (apunta hacia arriba), el ángulo es 0. Es decir, que la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo es $(0, 0, 1)$ (apunta hacia atrás).

Vamos a notar por $\psi_I : S^2 \to \mathbb{R}$ a la función que, a cada dirección del hombro izquierdo, le asigna el ángulo del que estamos hablando. Las suposiciones anteriores se reducen a:

$$\left\{\begin{array}{lcl} \psi_I\,(0,1,0) & = & \pi/4\,, \\ \psi_I\,(\cos\varphi, 0, \sin\varphi) & = & 0\,, \text{ para cada } \varphi \in [-\pi, 0]\,, \\ \psi_I\,(0,0,1) & = & -\pi/2\,, \\ \psi_I\,(0,-1,0) & = & 0\,. \end{array}\right.$$

Lo siguiente que vamos a tratar de entender es que, si fijamos un $\varphi \in [-\pi, 0]$, entonces debe cumplirse que

$$\lim_{\theta \to 0^+} \psi_I\,(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = -\,(\varphi + \pi/4)\,.$$

¿Por qué? En primer lugar, vemos que estamos usando coordenadas esféricas para representar los vectores de S^2. φ es el azimut y θ es la colatitud respecto al eje Y. Entonces, al tomar el límite cuando θ tiende a 0 por la derecha, estamos yendo hacia el vector $(0, 1, 0)$ por el arco de azimut constante φ. Observamos que, para todos los vectores de este arco cuya colatitud es $\theta \in (0, \pi/2)$, la proyección sobre el plano $y = 0$ del vector auxiliar del paso anterior tiene la misma dirección y sentido: $(\cos\varphi, 0, \sin\varphi)$.

Así pues, al calcular el límite anterior, tiene sentido que el resultado sea el ángulo entre el vector $(\cos\varphi, 0, \sin\varphi)$ y el de la dirección del antebrazo cuando el ángulo del codo es de 90 grados, la dirección del hombro es $(0,1,0)$ y la rotación del hombro es máxima. Teniendo en cuenta que $\psi_I(0,1,0) = \pi/4$ y el vector auxiliar de $(0,1,0)$ es $(0,0,-1)$ es fácil llegar al resultado final.

En otras palabras, el resultado del límite es el ángulo entre el límite de los vectores auxiliares y la dirección del antebrazo que debemos considerar.

Sea pues $\varphi \in [-\pi, 0]$. Consideremos la función $\psi_I^{\varphi} : [0, \pi/2] \to [-\pi, \pi)$ definida por $\psi_I^{\varphi}(\theta) = \psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi)$, para cada $\theta \neq 0$ y extendida de forma contínua a $\theta = 0$. Vamos a suponer que ψ_I^{φ} es lineal. Como $\psi_I^{\varphi}(0) = -(\varphi + \pi/4)$ y $\psi_I^{\varphi}(\pi/2) = 0$, tenemos que

$$\psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = \left(\varphi + \frac{\pi}{4}\right)\left(\frac{2}{\pi}\theta - 1\right), \text{ para cada } \theta \in (0, \pi/2]\,.$$

Siguiendo la misma idea, si fijamos $\varphi \in [-\pi, 0]$, debe cumplirse que

$$\lim_{\theta\to\pi^-} \psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = \varphi + \pi/2\,.$$

Y, de nuevo, suponiendo la linealidad de la función correspondiente, obtenemos que

$$\psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = \left(\varphi + \frac{\pi}{2}\right)\left(\frac{2}{\pi}\theta - 1\right), \text{ para cada } \theta \in [\pi/2, \pi)\,.$$

Hasta ahora hemos considerado las direcciones del hombro cuya tercera componente es no negativa. Ahora faltan las otras direcciones. Lo primero que debemos hacer es determinar el ángulo para todas las direcciones que son paralelas al suelo (estando erguido), es decir, aquellas para las que la segunda componente es nula. Vamos a definir la función $\psi_I' : [0, \pi) \to [-\pi, \pi)$, definida por $\psi_I'(\varphi) = \psi_I(\cos\varphi, 0, \sin\varphi)$ para cada φ. Suponemos de nuevo que esta función es lineal, así que, como $\psi_I'(0) = 0$ y $\psi_I'(\pi/2) = -\pi/2$, se tiene que $\psi_I(\cos\varphi, 0, \sin\varphi) = -\varphi$, para cada $\varphi \in [0, \pi)$.

Fijemos ahora $\varphi \in (0, \pi)$. Siguiendo la misma idea de antes, debe cumplirse que

$$\lim_{\theta\to 0^+} \psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = -(\varphi + \pi/4)\,.$$

Y, de nuevo, suponiendo la linealidad de la función correspondiente, obtenemos que

$$\psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = -\varphi - \frac{\pi}{4} + \frac{\theta}{2}, \text{ para cada } \theta \in [\pi/2, \pi)\,.$$

Por último, fijando de nuevo $\varphi \in (0, \pi)$, debe cumplirse que

$$\lim_{\theta\to\pi^-} \psi_I(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi) = \varphi + \pi/2\,.$$

```python
if np.isclose(y, 1):
    angle = pi / 4.
elif np.isclose(y, -1):
    angle = 0.
else:
    tita = colatitude(x, z, y)
    fi = azimuth(x, z, y)
    if z<=0:
        if y>=0:
            angle = (fi + pi / 4.) * (2. * tita / pi - 1.)
        else:
            angle = (fi + pi / 2.) * (2. * tita / pi - 1.)
    else:
        if y>=0:
            angle = tita / 2. - fi - pi / 4.
        else:
            angle = - fi + (2. * fi + pi / 2.) * (2. * tita / pi - 1.)
```

Figura 4: Código del ángulo de la función 'max_left_shoulder_rotation'.

Con lo que se sigue que, para cada $\theta \in [\pi/2, \pi)$,

$$\psi_I\left(\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi\right) = -\varphi + \left(2\varphi + \frac{\pi}{2}\right)\left(\frac{2}{\pi}\theta - 1\right).$$

En resumen, dado un vector v de S^2, consideramos $\theta \in [0, \pi]$ y $\varphi \in [-\pi, \pi)$ tales que $v = (\sin\theta\cos\varphi, \cos\theta, \sin\theta\sin\varphi)$ en la base del hombro izquierdo. Se tiene entonces que

$$\psi_I\left(v\right) = \begin{cases} \pi/4 & \text{si} \quad \theta = 0\,, \\ 0 & \text{si} \quad \theta = \pi\,, \\ \left(\varphi + \frac{\pi}{4}\right)\left(\frac{2}{\pi}\theta - 1\right) & \text{si} \quad \theta \in (0, \pi/2] \;\text{ y }\; \varphi \in [-\pi, 0]\,, \\ \left(\varphi + \frac{\pi}{2}\right)\left(\frac{2}{\pi}\theta - 1\right) & \text{si} \quad \theta \in [\pi/2, \pi] \;\text{ y }\; \varphi \in [-\pi, 0]\,, \\ -\varphi + \frac{\pi}{4}\left(\frac{2}{\pi}\theta - 1\right) & \text{si} \quad \theta \in (0, \pi/2] \;\text{ y }\; \varphi \in [0, \pi)\,, \\ -\varphi + \left(2\varphi + \frac{\pi}{2}\right)\left(\frac{2}{\pi}\theta - 1\right) & \text{si} \quad \theta \in (0, \pi/2] \;\text{ y }\; \varphi \in [0, \pi)\,. \end{cases}$$

En la Figura 4 podemos ver el código usado para calcular el ángulo en la función 'max_left_shoulder_rotation'.

Hasta aquí el brazo izquierdo. Para el derecho, tenemos que tener en cuenta que tercer vector de la base del hombro apunta hacia el frente, y no hacia atrás. Por otra parte, el ángulo lo consideraremos en sentido horario respecto a la dirección del hombro, justo al contrario que para el brazo izquierdo.

Sea $\psi_D : S^2 \to \mathbb{R}$ la función que asigna, a cada dirección del hombro derecho, el ángulo que estamos considerando. Las estimaciones de las que partimos son equivalentes a las de ψ_I:

$$\begin{cases} \psi_D\left(0,1,0\right) & = & \pi/4\,, \\ \psi_D\left(\cos\varphi,0,\sin\varphi\right) & = & 0\,,\ \text{para cada } \varphi\in[0,\pi]\,, \\ \psi_D\left(0,0,-1\right) & = & -\pi/2\,, \\ \psi_D\left(0,-1,0\right) & = & 0\,. \end{cases}$$

Y, siguiendo las ideas anteriores, llegamos a que, si v es un vector de S^2, y consideramos $\theta\in[0,\pi]$ y $\varphi\in(-\pi,\pi]$ tales que $v=(\sin\theta\cos\varphi,\cos\theta,\sin\theta\sin\varphi)$ en la base del hombro derecho, se tiene entonces que

$$\psi_D\left(v\right)=\begin{cases} \pi/4 & \text{si} \quad \theta=0\,, \\ 0 & \text{si} \quad \theta=\pi\,, \\ \left(\frac{\pi}{4}-\varphi\right)\left(\frac{2}{\pi}\theta-1\right) & \text{si} \quad \theta\in(0,\pi/2] \ \text{ y } \varphi\in[0,\pi]\,, \\ \left(\frac{\pi}{2}-\varphi\right)\left(\frac{2}{\pi}\theta-1\right) & \text{si} \quad \theta\in[\pi/2,\pi] \ \text{ y } \varphi\in[0,\pi]\,, \\ \varphi+\frac{\pi}{4}\left(\frac{2}{\pi}\theta-1\right) & \text{si} \quad \theta\in(0,\pi/2] \ \text{ y } \varphi\in(-\pi,0]\,, \\ \varphi+\left(\frac{\pi}{2}-2\varphi\right)\left(\frac{2}{\pi}\theta-1\right) & \text{si} \quad \theta\in(0,\pi/2] \ \text{ y } \varphi\in(-\pi,0]\,. \end{cases}$$

Es evidente que en estos razonamientos la cantidad de suposiciones no triviales han sido inmensas, así que es natural hacerse las siguientes preguntas: ¿Por qué? ¿Por qué esto debería funcionar? En primer lugar, por ahora realmente no sé si esto debería funcionar, aunque creo y espero que sí. En segundo lugar, debemos tener claro que el objetivo de estas funciones es ser aplicadas para el estudio del movimiento haciendo lengua de signos, por lo que las posiciones de los brazos no deberían estar en los límites del cuerpo humano. Probablemente, si aplicáramos estas funciones para estudiar los movimientos de una persona haciendo yoga, el resultado sería desastroso. La idea tras estas funciones es considerar, para cada dirección del hombro, una rotación de hombro de referencia, y esperar que ninguna de las posiciones que nos encontremos en la lengua de signos sobrepase esta rotación o justamente la opuesta. En resumen, no hay ninguna garantía de que estas funciones vayan a servir, pero considero que hay motivos por los que pensar que no deberían dar problemas.

Ya hemos completado los dos primeros pasos, los dos siguientes son prácticamente inmediatos. En primer lugar, vamos a calcular el vector 'aux_normal' como el vector normal al plano que contiene a la dirección del hombro y al vector auxiliar del primer paso. Teniendo en cuenta el sentido del ángulo que hemos considerado en el segundo paso (antihorario para el brazo izquierdo, horario para el brazo derecho), el vector 'aux_normal' va a ser el producto tensorial del vector auxiliar con la dirección del hombro para el brazo derecho, y el mismo producto pero en el otro orden para el brazo derecho. Una vez que tenemos este vector, para obtener la dirección del antebrazo cuando el ángulo del codo es de 90 grados y el hombro está rotado al máximo es símplemente

$$\cos\left(\text{angulo}\right)\text{aux\_vector}+\sin\left(\text{angulo}\right)\text{aux\_normal}\,.$$

El último paso consiste símplemente en hacer el producto tensorial entre la dirección del hombro y el vector recién calculado y devolver el resultado. En la Figura 5 vemos las líneas de la función 'max_left_shoulder_rotation'que realizan estos dos últimos pasos.

```
aux_normal = np.cross(aux_vector, left_shoulder_direction)
left_elbow_direction_max_rotation = np.cos(angle) * aux_vector + np.sin(angle) * aux_normal

return np.cross(left_shoulder_direction, left_elbow_direction_max_rotation)
```

Figura 5: Últimas líneas de la función 'max_left_shoulder_rotation'.