Índice general

CAPÍTULO 1

Modelo de cámara y estimación de pose monocular

1.1. Introducción

Se le llama "estimación de pose" al proceso mediente el cual se calcula en qué punto del mundo y con qué orientación se encuentra determinado objeto respecto de un eje de coordenadas previamente definido al que se lo llama "ejes del mundo". Las aplicaciones de realidad aumentada requieren de un modelado preciso del entorno respecto de estos ejes, para poder ubicar correctamente los agregados virtuales dentro del modelo y luego dibujarlos de forma coherente en la imagen vista por el usuario. El objeto cuya estimación de pose resulta de mayor importancia es la cámara, ya que por ésta es por donde se mira la escena y es respecto de ésta que los objetos virtuales deben ubicarse de manera consistente. Una forma de estimar la pose de la cámara es mediante el uso de las imágenes capturadas por ella misma. Asimismo, el concepto "monocular" hace referencia al uso de una sola cámara, ya que es posible trabajar con más de una.

Para poder obtener información relevante a partir de las imágenes tomadas por una cámara, resulta necesario contar con un modelo preciso de su arquitectura ya que no todas las cámaras son iguales. El modelo más comunmente utilizado es el denominado *pin-hole*. Para modelar completamente la arquitectura de la cámara se deben estimar ciertos "parámetros intrínsecos" a ésta, y eso se logra luego de realizados ciertos experimentos. A la estimación de estos parámetros se le denomina "calibración de la cámara".

En este capítulo se verá en detalle el modelo de cámara *pin-hole*, tomando en cuenta la distorsión introducida por las lentes. Más adelante, se mencionarán distintos métodos para la calibración de una cámara y se verá en detalle en particular, el método de Zhang.

También se presentan los algoritmos más utilizados para el problema de estimación de pose, entre ellos el DLT(Direct Linear Transform), P*n*P(Perspective *n* Point) y RANSAC(RANdom SAmple Consensus).

Finalmente se presentan las diferentes maneras que hay para representar los ángulos de la pose y los problemas que se presentan cuando se trabaja con estas representaciones.

1.2. Modelo de cámara *pin-hole* [?]

1.2.1. Fundamentos y definiciones

Este modelo consiste en un centro óptico O, en donde convergen todos los rayos de la proyección y un plano imagen en el cual la imagen es proyectada. Se define *distancia focal* (f) como la distancia entre el centro óptico O y la intersección del eje óptico con el plano imagen (punto C). Ver Figura 1.1.

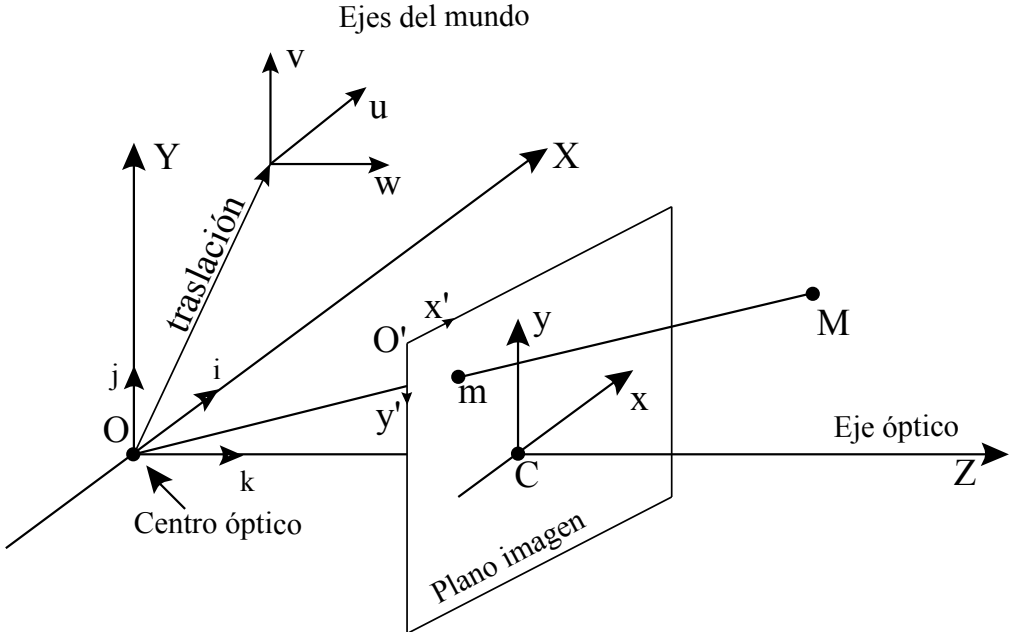


Figura 1.1: Modelo de cámara pin-hole.

Se llama proceso de pryección al proceso en el que se asocia al punto **M** del mundo, un punto **m** en la imagen. Para modelar el mismo es necesario referirse a varias transformaciones y varios ejes de coordenadas.

- *Coordenadas del mundo*: son las coordenadas que describen la posición 3D del punto **M** respecto de los ejes del mundo (u,v,w). La elección de los ejes del mundo es arbitraria.
- *Coordenadas de la cámara*: son las coordenadas que describen la posición del punto **M** respecto de los ejes de la cámara (X,Y,Z). i, j y k son los versores de este eje de coordenadas.
- *Coordenadas de la imagen*: son las coordenadas que describen la posición del punto 2D, **m** respecto del centro del plano imagen, C. Los ejes de este sistema de coordenadas son (x,y).
- *Coordenadas normalizadas de la imagen*: son las coordenadas que describen la posición del punto 2D, **m**, respecto del eje de coordenadas (x',y') situado en la esquina superior izquierda del plano imagen.

La transformación que lleva al punto **M**, expresado respecto de los ejes del mundo, al punto **m**, expresado respecto del sistema de coordenadas normalizadas de la imagen, se puede ver como la composición de dos transformaciones menores. La primera, es la que realiza la proyección que transforma a un punto definido respecto del sistema de coordenadas de la cámara (X,Y,Z) en otro punto sobre el plano imagen expresado respecto del sistema de coordenadas normalizadas de la

imagen (x', y'). Véase que una vez calculada esta transformación, es una constante característica de cada cámara. Al conjunto de valores que definen esta transformación, se le llama "parámetros intrínsecos" de la cámara. La segunda, es la transformación que lleva de expresar un punto respecto de los ejes del mundo (u, v, w), a ser expresado según los ejes de la cámara (X, Y, Z). Esta última transformación varía conforme se mueve la cámara (respecto de los ejes del mundo) y el conjunto de valores que la definen es denominado "parámetros extrínsecos" de la cámara. Del cálculo de estos parámetros es que se obtiene la estimación de la pose de la cámara.
De lo anterior se concluye rápidamente que si se le llama H a la matriz proyección total, tal que:

$$m = H.M,$$

entonces:

$$H = I.E$$

donde I corresponde a la matriz proyección asociada a los parámetros intrínsecos y E corresponde a la matriz asociada a los parámetros extrínsecos. Ambos juegos de parámetros acarrean información muy valiosa:

- **Parámetros extrínsecos:** pose de la cámara.
  - Traslación: ubicación del centro óptico de la cámara respecto de los ejes del mundo.
  - Rotación: rotación del sistema de coordenadas de la cámara (X, Y, Z), respecto de los ejes del mundo.
- **Parámetros intrínsecos:** parámetros propios de la cámara. Dependen de su geometría interna y de su óptica.
  - Punto principal (C = $[x'_C, y'_C]$): es el punto intersección entre el eje óptico y el plano imagen. Las coordenadas de este punto vienen dadas en píxeles y son expresadas respecto del sistema normalizado de la imagen.
  - Factores de conversión píxel-milímetros (d_x, d_y): indican el número de píxeles por milímetro que utiliza la cámara en las direcciones x e y respectivamente.
  - Distancia focal (f): distancia entre el centro óptico (**O**) y el punto principal (**C**). Su unidad es el milímetro.
  - Factor de proporción (s): indica la proporción entre las dimensiones horizontal y vertical de un píxel.

1.2.2. Matriz de proyección

En la sección anterior se vio que es posible hallar una "matriz de proyección" H que dependa tanto de los parámetros intrínsecos de la cámara como de sus parámetros extrínsecos:

$$m = H.M$$

donde **M** y **m** son los puntos ya definidos y vienen expresados en "coordenadas homogéneas". Por más información acerca de este tipo de coordenadas ver [**?**].
Para determinar la forma de la matriz de proyección se estudia cómo se relacionan las coordenadas de **M** con las coordenads de **m**; para hallar esta relación se debe analizar cada transformación, entre los sistemas de coordenadas mencionados con anterioridad, por separado.

- **Proyección 3D - 2D:** de las coordenadas homogéneas del punto **M** expresadas en el sistema de coordenadas de la cámara (X_0, Y_0, Z_0, T_0), a las coordenadas homogéneas del punto **m** expresadas en el sistema de coordenadas de la imagen (x_0, y_0, s_0):
  Se desprende de la imagen 1.1 y algo de trigonometría la siguiente relación entre las coordenadas en cuestión y la distancia focal (f):

$$\frac{f}{Z_0} = \frac{x_0}{X_0} = \frac{y_0}{Y_0}$$

  A partir de la relación anterior:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \frac{f}{Z_0} \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

  Expresado en forma matricial, en coordenadas homogéneas:

$$\begin{pmatrix} x_0 \\ y_0 \\ s_0 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix}$$

- **Transformación imagen - imagen:** de las coordenadas homogéneas del punto **m** expresadas respecto del sistema de coordenadas de la imagen (x_0, y_0, s_0), a las coordenadas homogéneas de él mismo pero expresadas respecto del sistema de coordenadas normalizadas de la imagen (x'_0, y'_0, s'_0):

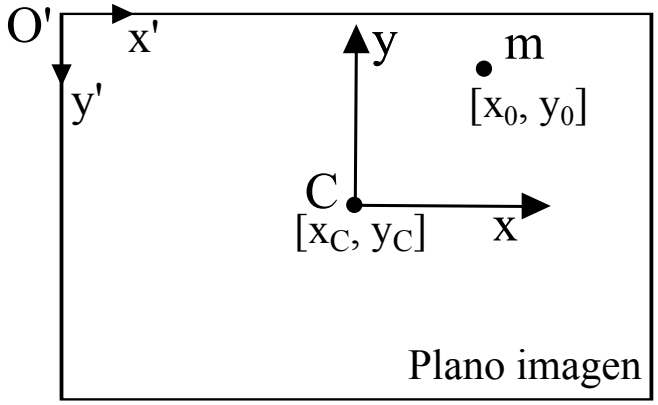


Figura 1.2: Relación entre el sistema de coordenadas de la imagen y el sistema de coordenandas normalizadas de la imagen.

  Se les suma, a las coordenadas de **m** respecto del sistema de la imagen, la posición del punto C respecto del sistema normalizado de la imagen (x'_C, y'_C). Las coordenadas de **m** dejan de ser expresadas en milímetros para ser expresadas en píxeles. Aparecen los factores de conversión d_x y d_y:

$$x'_0 = d_x.x_0 + x'_C$$
$$y'_0 = d_y.y_0 + y'_C$$

  Se obtiene entonces la siguiente relación matricial, en coordenadas homogéneas:

$$\begin{pmatrix} x'_0 \\ y'_0 \\ s'_0 \end{pmatrix} = \begin{pmatrix} d_x & 0 & x'_C \\ 0 & d_y & y'_C \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$$

- **Matriz de parámetros intrínsecos** (I)**:** de las coordenadas homogéneas del punto **M** expresadas en el sistema de coordenadas de la cámara $(X_0, Y_0, Z_0, 1)$, a las coordenadas homogéneas del punto **m** expresadas respecto del sistema de coordenadas normalizadas de la imagen $(x'_0 y'_0, s'_0)$:
  Se obtiene combinando las dos últimas transformaciones. Nótese que como ya se aclaró, depende únicamente de parámetros propios de la construcción de la cámara:

$$I = \begin{pmatrix} d_x.f & 0 & x'_C & 0 \\ 0 & d_y.f & y'_C & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

  Nota: De forma genérica se puede agregar a la matriz de parámetros intrínsecos del modelo *pin-hole* un parámetro **s** llamado en inglés *skew parameter*, o "parámetro de proporción" en Español. Este parámetro toma valores distintos de cero muy rara vez, pues modela los casos en los que los ejes x e y de los píxeles de la cámara no son perpendiculares entre sí. En casos realistas, $s \neq 0$ cuando por ejemplo se toma una fotografía de una fotografía. La matriz de parámetros intrínsecos, tomando en cuenta este parámetro, tendrá la forma:

$$I = \begin{pmatrix} d_x.f & s & x'_C & 0 \\ 0 & d_y.f & y'_C & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- **Matriz de parámetros extrínsecos** (E)**:** de las coordenadas homogéneas del punto **M** expresadas respecto del sistema de coordenadas del mundo (U_0, V_0, W_0, P_0), a las coordenadas homogéneas de él mismo pero expresadas respecto del sistema de coordenadas de la cámara (X_0, Y_0, Z_0, T_0):
  Se obtiene de estimar la pose de la cámara respecto de los ejes del mundo y es la combinación de, primero una rotación R_{3x3}, y luego una traslación T_{3x1}. Se obtiene entonces la siguiente representación matricial:

$$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ T_0 \end{pmatrix} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \end{pmatrix}$$

  donde la matriz de parámetros extrínsecos desarrollada toma la forma:

$$E = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Matriz de proyección** (H)**:** de las coordenadas homogéneas del punto **M** expresadas respecto del sistema de coordenadas del mundo (U_0, V_0, W_0, P_0), a las coordenadas homogéneas del punto **m** expresadas respecto del sistema de coordenadas normalizadas de la imagen (x'_0, y'_0, s'_0):
  Es la proyección total y se obtiene combinando las dos transformaciones anteriores:

$$\begin{pmatrix} x'_0 \\ y'_0 \\ s'_0 \end{pmatrix} = \begin{pmatrix} d_x.f & 0 & x'_C & 0 \\ 0 & d_y.f & y'_C & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \end{pmatrix}$$

1.3. Distorsión introducida por las lentes

Hasta el momento se asumió que el modelo lineal presentado para la proyección de cualquier punto del mundo en el plano imagen de la cámara es lo suficientemente preciso en todos los casos. Sin embargo, en casos reales, y cuando las lentes de las cámaras no son del todo buenas, la distorsión introducida por estas se hace notar. Dado el punto **M** de coordenadas (X_0, Y_0, Z_0) respecto de los ejes de la cámara, se le llama distorsión a la diferencia entre su proyección ideal en el plano imagen (x_0, y_0) y su proyección real $(\tilde{x_0}, \tilde{y_0})$. La más común de todas, es la denominada "distorsión radial", ya que su magnitud depende del radio medido desde el punto principal del plano imagen, hasta las coordenadas del punto en cuestión.

La forma de solucionar el presente problema es realizar una corrección de la distorsión, modelando a la misma de la siguiente manera:

$$\begin{pmatrix} \tilde{x_0} \\ \tilde{y_0} \end{pmatrix} = L(r).\begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

donde r es la distancia radial $\sqrt{x_0^2 + y_0^2}$ y $L(r)$ es un factor de distorsión que depende únicamente del radio r. Si se desarrolla la ecuación anterior, y se expresa en píxeles, respecto del sistema de coordenadas normalizadas de la imagen; se obtiene lo siguiente:

$$\tilde{u_0}' = x'_C + L(r)(x'_0 - x'_C)$$
$$\tilde{v_0}' = y'_C + L(r)(y'_0 - y'_C)$$

donde $(\tilde{x_0}', \tilde{y_0}')$ son las coordenadas reales de la proyección medidas en píxeles, (x'_0, y'_0) son las coordenadas ideales de la proyección medidas también en píxeles y (x'_C, y'_C) son las coordenadas del punto principal. Véase que en este caso $r = \sqrt{(x'_0 - x'_C)^2 + (y'_0 - y'_C)^2}$.

La función $L(r)$ es definida sólo para valores positivos de r y $L(0) = 1$. Una aproximación a la función arbitraria $L(r)$ puede ser una expansión de Taylor: $L(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + ...$ Finalmente, a la hora de calcular los parámetros intrínsecos de una cámara, también deben ser estimados sus coeficientes de distorsión radial $\{k_1, k_2, k_3, k_4, ...\}$.

1.4. Métodos para la calibración de cámara

Como se vio algunos párrafos atrás, el proceso mediante el cual se calculan los parámetros intrínsecos reales de una cámara es denominado "calibración de cámara". Existen varios métodos para calibrar una cámara; sin embargo, los tres algoritmos, basados en modelos planos, más ampliamente utilizados alrededor del mundo [**?**] son el método de Zhang [**?**], el método de R.Y. Tsai [**?**] y un método llamado "Direct Linear Transform" (DLT) [**?**] . Para calibrar las cámaras utilizadas en este proyecto, se trabajó con una implementación en *Matlab* basada en el método de Zhang ([**?**]), que afortunadamente dio resultados muy buenos. Por eso, se explicará a continuación, de forma breve, cómo funciona este método. Por dudas respecto de cualquier resultado matemático expuesto sin los cálculos intermedios, siempre se recomienda leer el artículo original.

El método de Zhang es muy sencillo y flexible. Sólo requiere de la cámara a calibrar, una computadora y una imagen patrón (plana), de tipo damero; a la que se le tomarán al menos dos fotografías desde orientaciones distintas. En la figura 1.3 se ve una de las imágenes utilizadas para

Figura 1.3: Imagen de un damero, utilizada para calibrar la cámara del *iPad* durante el proyecto.

calibrar la cámara del *iPad* durante el proyecto. Ni las posiciones de la cámara en cada caso, ni el movimiento entre estas posiciones tienen por qué ser conocidos. Este método devuelve los parámetros intrínsecos de la cámara correspondientes al modelo *pin-hole* visto anteriormente, sus parámetros extrínsecos para cada fotografía utilizada para la calibración y la distorsión radial de sus lentes.

Recuérdese que la relación entre un punto 3D **M** expresado respecto de los ejes de coordenadas del mundo y su proyección en el plano imagen **m**, expresada respecto de los ejes normalizados de la imagen, viene dada por:

$$m = I.E.M$$

donde E representa a la matriz de parámetros extrínsecos e I representa a la matriz de parámetros intrínsecos de la cámara. Además:

$$I = \begin{pmatrix} \alpha & s & x'_C \\ 0 & \beta & y'_C \\ 0 & 0 & 1 \end{pmatrix}$$

con $\alpha = d_x.f$ y $\beta = d_y.f$.

Se asume en este método que el sistema de coordenadas del mundo "reposa" sobre la imagen patrón; o lo que es lo mismo, que esta se encuentra en $Z = 0$. Se obtiene entonces la siguiente simplificación:

$$\begin{pmatrix} x'_0 \\ y'_0 \\ 1 \end{pmatrix} = I.\begin{pmatrix} r_1 & r_2 & r_3 & t \end{pmatrix}.\begin{pmatrix} U_0 \\ V_0 \\ W_0 \\ 1 \end{pmatrix} = I.\begin{pmatrix} r_1 & r_2 & t \end{pmatrix}.\begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix}$$

donde $(U_0, V_0, W_0, 1)^T$ denota las coordenadas homogéneas del punto **M** respecto de los ejes del mundo y $(x'_0, y'_0, 1)^T$ representa las coordenadas homogéneas de su proyección en el plano imagen, **m**, respecto de los ejes normalizados de la imagen. Se le llamó r_i a la i-ésima columna de la matriz rotación de los parámetros extrínsecos de la cámara.

Dada una fotografía de la imagen patrón plana (figura 1.3), es posible estimar una homografía que relacione a los puntos de la imagen con sus correspondientes en la fotografía. Si se toma en cuenta que dicha homografía vale $H = (h_1, h_2, h_3) = I.(r_1, r_2, t)$, con h_i la i-ésima columna de la matriz, y que las columnas r_1 y r_2 son ortonormales entre sí, realizando algo de matemática se llega

a que:

$$h_1^T.(I^{-1})^T.I^{-1}.h_2 = 0$$
$$h_1^T.(I^{-1})^T.I^{-1}.h_1 = h_2^T.(I^{-1})^T.I^{-1}.h_2$$

Las anteriores son las únicas dos relaciones básicas entre parámetros intrínsecos que se pueden obtener a partir de una única homografía. Esto es porque una homografía tiene 8 grados de libartad y existen 6 parámetros extrínsecos (3 para la traslación y 3 para la rotación).

Si se define la matriz B como sigue:

$$B = (I^{-1})^T.I^{-1} = \begin{pmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha^2} & -\frac{s}{\alpha^2.\beta} & \frac{s.v'_P - u'_P.\beta}{\alpha^2.\beta} \\ -\frac{s}{\alpha^2.\beta} & \frac{s^2}{\alpha^2.\beta^2} + \frac{1}{\beta^2} & -\frac{s(s.v'_P - u'_P.\beta)}{\alpha^2.\beta^2} - \frac{v'_P}{\beta^2} \\ \frac{s.v'_P - u'_P.\beta}{\alpha^2.\beta} & -\frac{s(s.v'_P - u'_P.\beta)}{\alpha^2.\beta^2} - \frac{v'_P}{\beta^2} & \frac{(s.v'_P - u'_P.\beta)^2}{\alpha^2.\beta^2} + \frac{v'^2_P}{\beta^2} + 1 \end{pmatrix}$$

se ve fácilmente que esta es simétrica, por lo que quedará absolutamente definida por un vector de 6 dimensiones:

$$b = (B11, B_{12}, B_{22}, B_{13}, B_{23}, B_{33})^T$$

Si además se define el vector variable v_{ij} de la siguiente manera:

$$v_{ij} = (h_{i1}.h_{j1}, h_{i1}.h_{j2} + h_{i2}.h_{j1}, h_{i2}.h_{j2}, h_{i3}.h_{j1} + h_{i1}.h_{j3}, h_{i3}.h_{j2} + h_{i2}.h_{j3}, h_{i3}.h_{j3})^T,$$

se tiene que:

$$h_i^T.B.h_j = V_{ij}^T.b$$

Las dos relaciones básicas entre parámetros intrínsecos obtenidas de una única homografía, vistas anteriormente, pueden ser reescritas como:

$$\begin{pmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{pmatrix}.b = V.b = 0$$

Utilizando n fotografías distintas de la imagen patrón, y por lo tanto n homografías distintas se obtiene una matriz V de tamaño $2.n \times 6$. Es sabido que si $n \geq 3$, el sistema matricial anterior tendrá una solución b única, que varía según cierto factor de escala. Sin embargo, si $n = 2$, es posible imponer la condición $s = 0$ y así también calcular al vector b de forma única, sin mayores problemas.

Una vez estimado b es posible recontruir la matriz de parámetros intrínsecos I, para luego utilizando I y las homografías H obtener los parámetros extrínsecos de la cámara para cada fotografía utilizada para la calibración.

El artículo de Zhang afirma que la solución obtenida hasta el momento no es del todo buena, pues se obtuvo minimizando una distancia algebraica y eso no tiene mucho sentido. Lo que se hace entonces es, utilizando las n fotografías tomadas para la calibración y los k puntos seleccionados en cada una de ellas, minimizar la siguiente ecuación:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \|m_{ij} - \hat{m}(I, E_i, M_j)\|^2$$

donde $\hat{m}(I, E_i, M_j)$ es la proyección del punto M_j en la imagen i utilizando la homografíau $H_i = I.E_i$. El resultado de dicha minimización no lineal será el resultado final. Este método requiere de valores

inicales para I y para los $E_i|_{i=1..n}$; que serán los obtenidos en los cálculos anteriores.

Finalmente se realiza una estimación de la distorsión radial utilizando un modelo muy similar al visto en la sección 1.3. Cabe destacar que cuando se estimaron los parámetros intrínsecos de las cámaras utilizadas en este proyecto, la distorsión radial no se tomó en cuenta y aún así los resultados obtenidos fueron realmente muy precisos.

1.5. Problema de estimación de pose

Como se mencionó en 1.2.1 la matriz de parámetros extrínsecos E representa a la pose de la cámara. El problema de estimación de pose consiste en determinar esta matriz dadas n correspondencias entre puntos M_i en el mundo 3D y puntos m_i en la imagen.

Existen varios algoritmos de estimación de pose, a continuación se presentan algunos.

1.5.1. *DLT*(*D*irect *L*inear *T*ransform)

Este método sirve para calcular la matriz $\mathbf{H}$ en la cual están implícitos los parámetros intrínsecos y extrínsecos. Si se conocen los parámetros intrínsecos se pueden despejar la matriz $\mathbf{E}$ con la información de la pose.

Como se vio anteriormente

$$\begin{pmatrix} x_i \\ y_i \\ s_i \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} & \mathbf{H}_{14} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} & \mathbf{H}_{24} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \mathbf{H}_{33} & \mathbf{H}_{34} \\ \mathbf{H}_{41} & \mathbf{H}_{42} & \mathbf{H}_{43} & \mathbf{H}_{44} \end{pmatrix} \cdot \begin{pmatrix} U_i \\ V_i \\ W_i \\ P_i \end{pmatrix}$$

Por comodidad a partir de ahora cuando se refiera a puntos 2D (x_i, y_i) serán expresados siempre desde el eje de coordenadas normalizadas de la imagen. Se debe notar que la matriz $\mathbf{H}$ puede ser multiplicada por un factor distinto de cero sin alterar el resultado de la proyección, esto se debe a que se trabaja con coordenadas homogéneas. Por lo tanto lo que define la proyección no son los elementos de $\mathbf{H}$ sino la relación entre todos los elementos (excepto el de factor de escala) y el elemento que da el factor de escala. Así entonces $\mathbf{H}$ tiene 12 elementos pero solamente 11 grados de libertad.

Cada correspondencia $M_i \leftrightarrow m_i$ aporta dos ecuaciones linealmente independientes con los elementos de la matriz $\mathbf{H}$, $\mathbf{H}_{ij}$ como variables

$$\frac{\mathbf{H}_{11}X_i + \mathbf{H}_{12}Y_i + \mathbf{H}_{13}Z_i + \mathbf{H}_{14}}{\mathbf{H}_{31}X_i + \mathbf{H}_{32}Y_i + \mathbf{H}_{33}Z_i + \mathbf{H}_{34}} = x_i,$$

$$\frac{\mathbf{H}_{21}X_i + \mathbf{H}_{22}Y_i + \mathbf{H}_{23}Z_i + \mathbf{H}_{24}}{\mathbf{H}_{31}X_i + \mathbf{H}_{32}Y_i + \mathbf{H}_{33}Z_i + \mathbf{H}_{34}} = y_i$$

La ecuación para s_i se puede obtener como combinación lineal de las de x_i y y_i, por eso no se tiene en cuenta. Estas ecuaciones se pueden reescribir como $\mathbf{A}h = 0$ donde

$$h = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} & \mathbf{H}_{14} & \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} & \mathbf{H}_{24} & \mathbf{H}_{31} & \mathbf{H}_{32} & \mathbf{H}_{33} & \mathbf{H}_{34} \end{pmatrix}^T$$

y

$$\mathbf{A} = \begin{pmatrix} X_0 & Y_0 & Z_0 & 1 & 0 & 0 & 0 & 0 & -x_0X_0 & -x_0Y_0 & -x_0Z_0 & -x_0 \\ 0 & 0 & 0 & 0 & X_0 & Y_0 & Z_0 & 1 & -y_0X_0 & -y_0Y_0 & -y_0Z_0 & -y_0 \\ X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n-1} & Y_{n-1} & Z_{n-1} & 1 & 0 & 0 & 0 & 0 & -x_{n-1}X_{n-1} & -x_{n-1}Y_{n-1} & -x_{n-1}Z_{n-1} & -x_{n-1} \\ 0 & 0 & 0 & 0 & X_{n-1} & Y_{n-1} & Z_{n-1} & 1 & -y_{n-1}X_{n-1} & -y_{n-1}Y_{n-1} & -y_{n-1}Z_{n-1} & -y_{n-1} \end{pmatrix}$$

La transformación obtenida la matriz **H** tiene 11 grados de libertad como mencionó anteriormente, por lo tanto el rango de la matriz **A** es 11. Se realiza la descomposición SVD de la matriz **A**, el vector propio del valor singular de **A** con valor menor es la base del núcleo de **A**. Entonces se tiene que h es este vector a menos de una constante. Una vez que se tiene h se puede armar la matriz **H**. Luego se tiene que

$$\mathbf{E} = \mathbf{I}^{-1}\mathbf{H}$$

1.5.2. *PnP* (*P*erspective-*n*-*P*oint)

Si se cuenta con los parámetros intrínsecos, se puede utilizar un enfoque que se centre en calcular solamente la pose de la cámara. Dependiendo de la cantidad de correspondencias que se tienen entre la imagen y el modelo es posible obtener un número finito de soluciones de la pose. Si se tienen 1 o 2 correspondencias el problema tiene infinitas soluciones. Si se tienen 3 correspondencias(P3P) se obtienen hasta 4 posibles soluciones. Para 4 o más correspondencias se obtiene una única solución, siempre que los puntos no estén alineados. La idea detrás de este algoritmo es la siguiente:

- A partir de los puntos de la imagen m_i y conociendo la distancia focal f es posible calcular los versores j_i.

$$j_i = \frac{1}{\sqrt{x_i^2 + y_i^2 + f^2}} \begin{pmatrix} x_i \\ y_i \\ f \end{pmatrix}$$

- Con estos versores es posible determinar los ángulos que forman las lineas de vista de los puntos $\mathbf{M}_i$ entre sí.
- Se busca estimar las distancias $l_i = \|\mathbf{OM}_i\|$ entre el centro de la cámara y los puntos 3D $\mathbf{M}_i$ a partir de las relaciones dadas por los triángulos $\mathbf{OM}_i\mathbf{M}_j$.
- Una vez que se calculan las distancias l_i, los puntos $\mathbf{M}_i$ se expresan en el sistema de coordenadas de la cámara como $\mathbf{M}_i^{\mathbf{C}}$.
- Finalmente **R** y **T** quedan determinadas como la transformación que lleva puntos en el sistema de coordenadas del mundo a el sistema de coordenadas de la cámara.

En la bibliografía [**?**] y [**?**] se encuentran varios métodos para resolver numéricamente el problema.

1.5.3. RANSAC(RANdom SAmple Consensus)

Este es un algoritmo iterativo utilizado para estimar los parámetros de un modelo matemático de un conjunto de datos que contiene *outliers* (datos fuera del modelo). En particular se puede utilizar para el problema de estimación de pose cuando no se tienen las correspondencias entre puntos detectados y puntos del modelo.

A continuación se presenta el algoritmo:

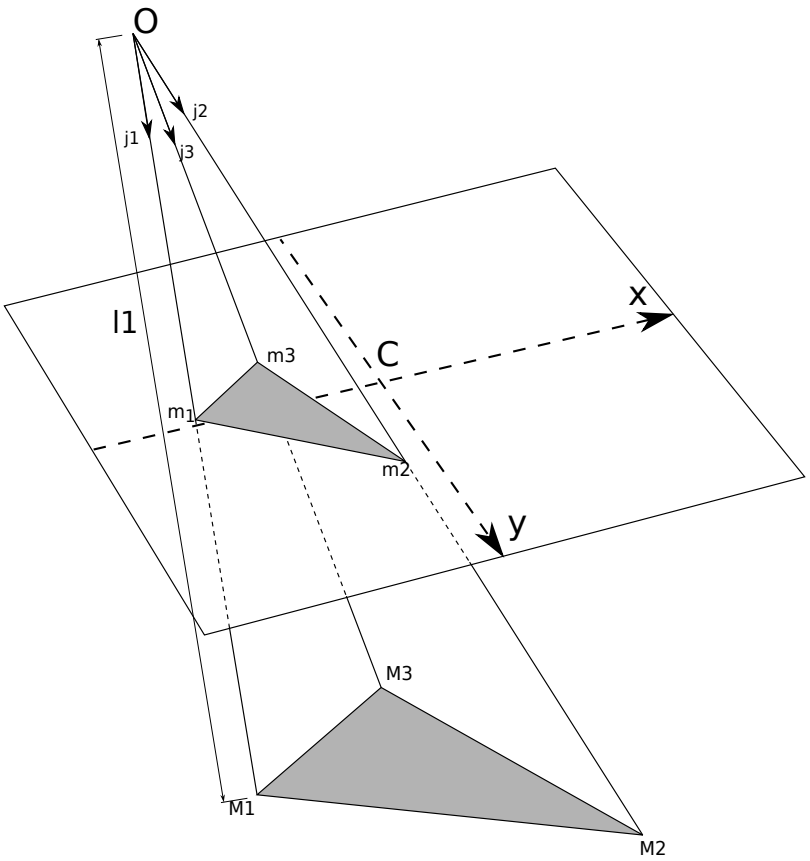


Figura 1.4: Geometría del problema P3P. Se busca calcular la distancia entre el centro óptico O y los puntos del modelo 3D

(1) Dado un modelo que requiere un mínimo de n puntos para determinar sus parámetros, y un conjunto de datos **P** tal que el número de puntos en **P** es mayor que n, se sortea un subconjunto S_1 de n puntos de **P** para instanciar el modelo. Con el modelo instanciado $\mathbf{M}_1$ se determina el subconjunto de decisión S_1^* de puntos de **P** que están a menos de una distancia t de $\mathbf{M}_1$.

(2) Si la cantidad de puntos en S_1^* es mayor que un umbral **T** entonces se elige el subconjunto de decisión S_1^* para computar el nuevo modelo $\mathbf{M}_1^*$.

(3) Si la cantidad de puntos en S_1^* es menor que **T**, se sortea un nuevo subconjunto S_2 y se repite el proceso. Si luego de una cantidad de N número de pruebas no se obtiene un subconjunto de decisión que cumple con el umbral **T**, se resuelve el modelo con el subconjunto de decisión mas grande obtenido, o se termina sin devolver modelo.

Los parámetros t, **T** y N, se eligen en base al modelo a estimar y a la probabilidad de encontrar un *outlier* en el conjunto de datos .

1.5.4. POSIT

Este es un algoritmo iterativo que se basa en utilizar la proyección ortogonal escalada (SOP) para resolver el problema de estimación de pose. Se necesita tener más de cuatro correspondencias entre puntos del modelo M_i y puntos en la imagen m_i. De todos los algoritmos presentados este fue el que se decidió utilizar. El desarrollo de la teoría de este algoritmo se encuentra en **??**. Este algoritmo tiene diferentes variantes. Por un lado esta la versión original del algoritmo y una versión que

resuelve el caso en que todos los puntos del modelo están en un mismo plano, (POSIT Coplanar). Luego se tiene una variante llamada SoftPOSIT que resuelve la estimación de pose sin la necesidad de conocer las correspondencias entre puntos del modelo 3D y puntos de la imagen en el caso en que los puntos del modelo no sean coplanares. Finalmente se tiene una variante de SoftPOSIT que trabaja con líneas.

La variante de SoftPOSIT de líneas fue el principal argumento para tomar la decisión ya que el detector de características que se usa es el LSD refch: lsd. Esta variante fue implementada sin éxito, pero en busca de esta implementación se desarrolló una versión de POSIT para puntos coplanares que no está presentada en la bibliografía y dio buenos resultados. Otro argumento a favor de esta opción es que se contaba con implemetaciones de algunas variantes. De [**?**] se obtuvieron las implementaciones de POSIT y POSIT coplanar en C y la implementación en MatLab de SoftPOSIT. Para la variante de SoftPOSIT de líneas sólo se contó con el artículo[**?**].

1.6. Representación de la pose de la cámara

Como se vio anteriormente, la pose de la cámara queda determinada por una matriz de rotación **R** y un vector de traslación **T**. La matriz **R** indica la orientación de la cámara respecto al mundo. Hay varias maneras de representar esta orientación, entre ellas se encuentran la representación matricial, la representación en ángulos de Euler y los *quaternions*. Dependiendo de la aplicación puede resultar más útil utilizar las diferentes representaciones.

1.6.1. Representación matricial

Esta representación es la que se introdujo en 1.2.2. En esta matriz las filas corresponden a los versores del sistema de coordenadas de la cámara expresados en las coordenadas del mundo. Se puede expresar como

$$\mathbf{R} = \begin{pmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{pmatrix}$$

La ventaja que tiene esta representación es que el pasaje de puntos en coordenadas del mundo a coordenadas de la cámara es directo, simplemente se multiplica por la matriz **R** al punto en coordenadas del mundo y se le suma el vector de traslación.

1.6.2. Ángulos de Euler

La matriz de rotación **R** se puede escribir como un producto de matrices que representan las rotaciones alrededor de los ejes x, y y z. No hay ninguna convención establecida en cuanto al orden en que se realizan las rotaciones. Por ejemplo si se toman ψ, θ y ϕ como los ángulos de rotación en torno a x, y y z respectivamente se tiene

$$\mathbf{R} = R_z(\phi)R_y(\theta)R_x(\psi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}$$

Desarrollando el producto se tiene que

$$\mathbf{R} = \begin{pmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{pmatrix}$$

1.6.2.1. Orden de rotaciones

Cuando se trabaja con matrices de rotaciones y ángulos de Euler es necesario saber en qué orden se aplican las rotaciones, pues no es una transformación conmutativa. Un objeto rotado primero según el eje x y luego según el eje y termina en una posición diferente que si se lo rota primero según y y luego según x. Esto se puede ver en la Figura 1.6.2.1.

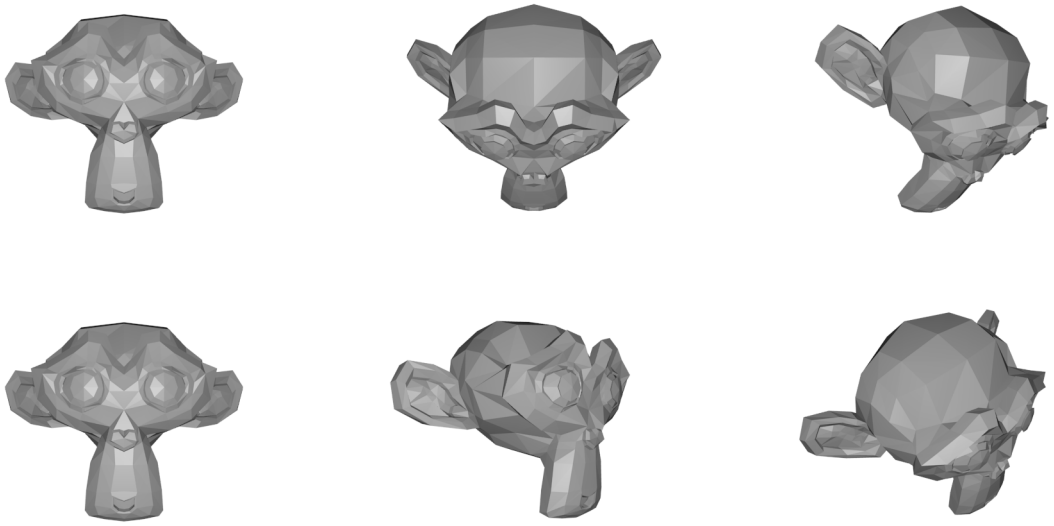


Figura 1.5: Se parte del mismo objeto, en la fila superior se aplica una rotación de 45° según x y luego una rotación de 45° según y. En la fila de abajo se aplican las mismas rotaciones pero en orden inverso. Se puede ver que se obtienen diferentes posiciones.

1.6.2.2. Cálculo de los ángulos de Euler

Si se tiene la matriz $\mathbf{R}$ es posible realizar la descomposición y obtener los ángulos de Euler. De $\mathbf{R}_{31}$ se obtiene el valor de θ

$$\theta = -\arcsin(\mathbf{R}_{31}).$$

Como $\sin(\theta) = \sin(\pi - \theta)$, puede haber dos posibles valores de θ(si $\mathbf{R}_{31} \neq \pm 1$)

$$\begin{aligned} \theta_1 &= -\arcsin(\mathbf{R}_{31}) \\ \theta_2 &= \pi - \theta_1 = \pi + \arcsin(\mathbf{R}_{31}) \end{aligned} \tag{1.1}$$

A partir de estos valores de $theta$ es posible encontrar dos juegos de ángulos que dan la misma matriz $\mathbf{R}$. Para calcular ψ se observa que

$$\frac{\mathbf{R}_{32}}{\mathbf{R}_{33}} = \tan\psi$$

de donde se deduce que

$$\psi = \arctan\left(\frac{\mathbf{R}_{32}}{\mathbf{R}_{33}}\right)$$

Es importante obtener el cuadrante al que pertenece el ángulo, por esto es que se usa la función arctan2 que esta disponible en C, recordar que la imagen de arctan es $[-\pi/2, \pi/2]$ por lo que

hay 2 cuadrantes que no se consideran. Como en los términos $\mathbf{R}_{32}$ y $\mathbf{R}_{33}$ aparece el término $\cos\theta$ multiplicando, hay que tener en cuenta su signo para obtener el valor de ψ. Si $\cos(\theta) > 0$, se tiene que $\psi = \arctan(\mathbf{R}_{32}/\mathbf{R}_{33})$. Si $\cos(\theta) < 0$, $\psi = \arctan(-\mathbf{R}_{32}/-\mathbf{R}_{33})$. Para tener en cuenta esto se toma

$$\psi = \arctan\left(\frac{\mathbf{R}_{32}/\cos\theta}{\mathbf{R}_{33}/\cos\theta}\right)$$

Por lo tanto los dos posibles valores para ψ son

$$\begin{aligned}\psi_1 &= \arctan\left(\frac{\mathbf{R}_{32}/\cos\theta_1}{\mathbf{R}_{33}/\cos\theta_1}\right)\\ \psi_2 &= \arctan\left(\frac{\mathbf{R}_{32}/\cos\theta_2}{\mathbf{R}_{33}/\cos\theta_2}\right)\end{aligned} \tag{1.2}$$

De manera similar se puede obtener ϕ. Se observa que

$$\frac{\mathbf{R}_{21}}{\mathbf{R}_{11}} = \tan\phi$$

por lo tanto se llega a

$$\begin{aligned}\phi_1 &= \arctan\left(\frac{\mathbf{R}_{21}/\cos\theta_1}{\mathbf{R}_{11}/\cos\theta_1}\right)\\ \phi_2 &= \arctan\left(\frac{\mathbf{R}_{21}/\cos\theta_2}{\mathbf{R}_{11}/\cos\theta_2}\right)\end{aligned} \tag{1.3}$$

Las ecuaciones 1.2 y 1.3 son válidas para el caso en que $\cos\theta \neq 0$

En el caso en que $\cos\theta = 0$ se tiene que $\theta = \pm\pi/2$, ademas los términos $\mathbf{R}_{11}$, $\mathbf{R}_{21}$, $\mathbf{R}_{32}$ y $\mathbf{R}_{33}$ son nulos. Por lo tanto se utilizan otros elementos de la matriz de rotación para hallar los ángulos restantes. En el caso en que $\theta = \pi/2$ se tiene que

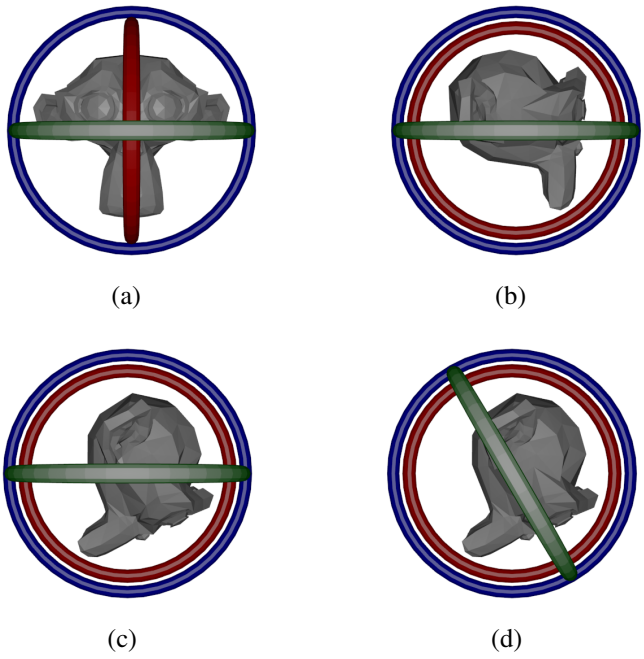
$$\begin{aligned}\mathbf{R}_{12} &= \sin\psi\cos\phi - \cos\psi\cos\phi = \sin(\psi-\phi)\\ \mathbf{R}_{13} &= \cos\psi\cos\phi + \sin\psi\sin\phi = \cos(\psi-\phi)\\ \mathbf{R}_{22} &= \sin\psi\sin\phi + \cos\psi\cos\phi = \cos(\psi-\phi) = \mathbf{R}_{13}\\ \mathbf{R}_{23} &= \cos\psi\sin\phi - \sin\psi\cos\phi = -\sin(\psi-\phi) = -\mathbf{R}_{12}\end{aligned}$$

Cualquier ψ y ϕ que verifiquen estas ecuaciones serán soluciones válidas. Usando las ecuaciones para $\mathbf{R}_{12}$ y $\mathbf{R}_{13}$ se tiene que

$$\begin{aligned}(\psi-\phi) &= \arctan(\mathbf{R}_{12}/\mathbf{R}_{13})\\ \psi &= \phi + \arctan(\mathbf{R}_{12}/\mathbf{R}_{13})\end{aligned}$$

Para el caso en que $\theta = -\pi/2$ se procede de igual manera y se llega a que

$$\begin{aligned}(\psi+\phi) &= \arctan(-\mathbf{R}_{12}/-\mathbf{R}_{13})\\ \psi &= -\phi + \arctan(-\mathbf{R}_{12}/-\mathbf{R}_{13})\end{aligned}$$



(a) (b)

(c) (d)

Figura 1.6: Los ejes rojo, verde y azul, son los ejes x, y y z. En (a) se ve la posición inicial. En (b) se puede ver el eje y rotado 90°. En (c) se ve la rotación según x de 60° respecto a la posición de (b). En (d) se ve la rotación según z de $-60°$ respecto a la posición de (b).

1.6.2.3. Gimbal lock

La gran desventaja que presenta la representación de la rotación mediante ángulos de Euler, es el problema denominado *gimbal lock*. Este problema se da cuando 2 de los ejes de rotación quedan alineados. Si hay dos ejes alineados, se pierde un grado de libertad ya que los dos ejes rotan de la misma manera. Para la composición de rotaciones que se utilizan en la aplicación el *gimbal lock* se da cuando se gira $\pi/2$ según y. Como se vio anteriormente, en el caso en que $\theta = \pi/2$ la matriz de rotación queda

$$\mathbf{R} = \begin{pmatrix} 0 & \sin(\psi-\phi) & \cos(\psi-\phi) \\ 0 & \cos(\psi-\phi) & -\sin(\psi-\phi) \\ -1 & 0 & 0 \end{pmatrix}$$

Esto es una rotación entorno al vector $(0,0,-1)$ de un ángulo $\alpha = \psi - \phi$

Esto se puede ver gráficamente en la Figura 1.6.2.3. Se realiza la rotación según y y se pueden ver que los ejes de x y z quedan alineados. Luego se realiza una rotación de 60° en torno a x y por otra parte también se hace otra igual pero de signo opuesto en torno a z. Se ve que la posición final, partiendo de los ejes alineados para una y otra rotacioón del modelo es la misma. Lo que cambia es la posición del eje y. Cuando se rota en torno a x, el eje y queda quieto porque está más arriba en la jerarquía de rotaciones para este caso particular. Cuando se rota en torno a z, el eje y se mueve ya que está por debajo de z en la jerarquía.

1.6.3. Cuaternios

Los cuaternios son una extensión a los números reales, son generados añadiendo las unidades imaginarias i, j y k. Se cumple que $i^2 = j^2 = k^2 = -1$. Un número cuaternio q se expresa como $q = a + bi + cj + dk$. También puede ser expresado como un escalar y un vector de 3 elementos $(a, \mathbf{v})$. Una rotación alrededor del versor ω un ángulo θ se puede expresar como el número cuaternio

unidad

$$q = \left(\cos\left(\frac{1}{2}\theta\right), \omega \sin\left(\frac{1}{2}\theta\right)\right)$$

Para rotar un punto 3D $\mathbf{M}$, se representa como un cuaternio $p = (0, \mathbf{M})$ y el punto p' rotado se calcula como

$$p' = qp\overline{q}.$$

El producto que se utiliza es el producto de cuaternios y $\overline{q} = \left(\cos\left(\frac{1}{2}\theta\right), -\omega \sin\left(\frac{1}{2}\theta\right)\right)$ es el cuaternio conjugado de q.

Esta representación evita el problema del *gimbal lock* pero tiene como contra que tiene un mayor costo computacional.