

Una forma de medir distancias 3D a partir de una imagen 2D sin saber la profundidad o disponer de un patrón conocido.

Abel Amado Gonzalez Bernad

Diciembre 2023

Resumen

Se ofrece un método para coordinar y medir elementos fotografiados con una cámara 2D sin información alguna de un patrón medido o conocimiento de la profundidad de los elementos en ella presentes. Es necesario restringir la presencia de los elementos fotografiados a un plano y conocer la posición extrínseca de la cámara respecto al mismo. Se ofrece un ejemplo implementación de uso para posicionar personas con YOLOv8 como detector de instancias y la medición del área de un folio en varias posiciones.

Código disponible en: <https://github.com/abelBEDOYA/plane-measurement>

1. Introducción

Es de interés el poder posicionar y medir elementos de una imagen 2D. La toma de esta imagen supone proyectar el espacio 3D a un sensor 2D, implicando una perdida de información irreversible. Por ello, típicamente para medir con precisión es común el uso de cámaras telecentricas, aunque tienen sus limitaciones (espacio de trabajo e iluminación). Para poder estimar posiciones con cualquier cámara 2D más allá de unas medidas en píxeles es posible contar con cierto aporte de información extra que permita compensar aquella perdida en la proyección.

- Conocer la profundidad del elemento fotografiado. Conociendo la distancia a la cámara paralela al eje óptico es posible coordinar su posición.
- Disponer y reconocer un patrón medido previamente. Conocer las dimensiones de aquello que se fotografía permite ubicar en posición y orientación dicho cuerpo extenso en el espacio 3D mediante algoritmos basados en *keypoints* y estimacion de pose [1] [2] [3].
- Limitar los elementos de la imagen a un plano y conocer la posición extrínseca de la cámara respecto a este. Esta restricción permite salvar la perdida de información debida a la proyección. Un método para coordinar y medir en esta condición es ofrecido aquí.

2. Modelización

Se considera el modelo de cámara pinhole, el cual, propone una cámara ideal que supone que todos los rayos que

forman la imagen pasan por el mismo punto. Este modeliza la óptica de la cámara en una matriz, Ec. 1 y una compensación no lineal de la posible deformación de lente que proponga la óptica de la cámara, Ec. 2 y Ec. 3. Estos son los coeficientes de los polinomios, $(k_1, k_2, p_1, p_2, k_3)$, que realizan una desplazamiento radial y tangencial de los píxeles de una imagen [4].

$$\mathbf{M} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{aligned} {}^r x_{\text{distorted}} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ {}^r y_{\text{distorted}} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (2)$$

$$\begin{aligned} {}^t x_{\text{distorted}} &= x + (2p_1 xy + p_2(r^2 + 2x^2)) \\ {}^t y_{\text{distorted}} &= y + (p_1(r^2 + 2y^2) + 2p_2 xy) \end{aligned} \quad (3)$$

2.1. Calibración de cámara

Cada cámara tiene una óptica y sensor distinto. Esto puede resumirse con el modelo cámara pinhole en la matriz de calibración y los coeficientes de distorsión. Estos son parámetros a ajustar y permiten la caracterización de la cámara. Para ello puede usarse el método de `cv2.cameraCalibration()` [4]. Este método debe recibir las dimensiones de un patrón conocido (tablero de ajedrez por ejemplo) y su posición en píxeles correspondiente a varias imágenes de ejemplo (imágenes de la calibración) tomadas desde distintas posiciones, Figura 1.

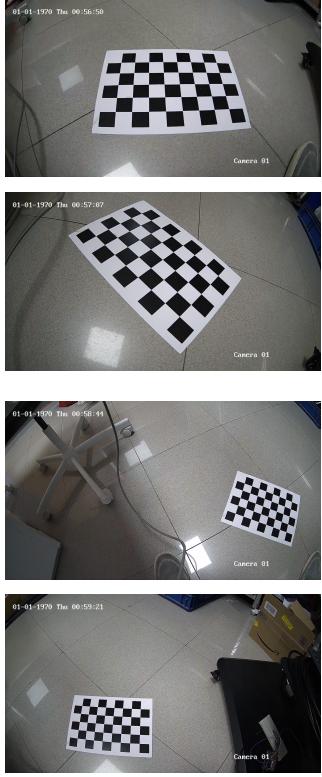


Figura 1: Imágenes de ejemplo del patrón de calibración para el ajuste de los parámetros de cámara.

2.2. Compensación de deformación

La calibración ofrece permite, entre otras cosas, eliminar la deformación de lente de la imagen, Figura 3.

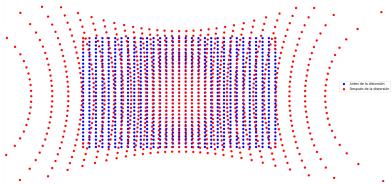


Figura 2: Índices de los píxeles de la imagen original y transformados de acuerdo a la deformación de lente. Se acentúa a medida que se alejan del centro.

Eliminar la distorsión consiste en transformar la posición de los píxeles, \mathbf{r}_p , a donde habrían estado si no hubiese deformación, \mathbf{r}'_p , es decir, realizar un mapeo de los índices de la matriz imagen.

2.3. De la realidad al sensor

La calibración de la cámara permite estimar a qué píxel de la imagen, $\mathbf{r}'_p = (x'_p, y'_p)^T$, se corresponderá cierta posición de la realidad tridimensional, $\mathbf{r} = (x, y, z)^T$ se fotografie, Ec. 4.

$$\mathbf{r}'_p = M \cdot \mathbf{r} \div z$$

esto es,

$$\begin{bmatrix} x'_p \\ y'_p \\ 1 \end{bmatrix} = \left(\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) \div z \quad (4)$$

Se trata de una transformación lineal dando lugar a los índices del píxel asociado a ese punto tridimensional. Estos píxeles deberán posteriormente recibir la compensación dada por la aberración de lente, Ec. 2 y Ec. 3.

2.4. Del sensor a la realidad, conocida la profundidad

Se trata de calcular inverso a llevar la realidad al sensor, es decir, partir del índice de píxel 2D y estimar el punto 3D es imposible de calcular debido la perdida de información en la proyección. Sin embargo, conociendo la profundidad, es decir, el valor de la componente z (dirección paralela al eje óptico) es posible coordinar en la realidad el punto identificando en la imagen. Así pues, invirtiendo la matriz de transformación M y despejando se tiene,

$$\mathbf{r} = M^{-1} \cdot \mathbf{r}'_p \cdot z$$

esto es,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left(\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x'_p \\ y'_p \\ 1 \end{bmatrix} \right) \cdot z \quad (5)$$

donde r' indica los índices del pixel desdistorsionado. Notar de nuevo las coordenadas homogéneas $r' = (x_p, y_p, 1)^T$

3. Cálculo cámara-plano

Ahora bien, conocida la modelización de la cámara pinhole y un método para coordinar posiciones 3D a partir de índices 2D sobre la imagen conocida la componente, z , aquí se propone un método para hacer lo mismo desconociendo dicha profundidad a los puntos.

Para ello, se debe exigir que todos los elementos que se pretende ubicar están contenidos en un plano, véase el suelo, el mar, un lago, un campo, ... Por simplicidad, este será aquel definido por la ecuación $z = 0$. Debe ser conocida la posición extrínseca de la cámara respecto al plano, esto es,

- Su altura, h .
- Ángulo de incidencia, β
- Giro de la cámara sobre su propio eje (*roll*), γ .

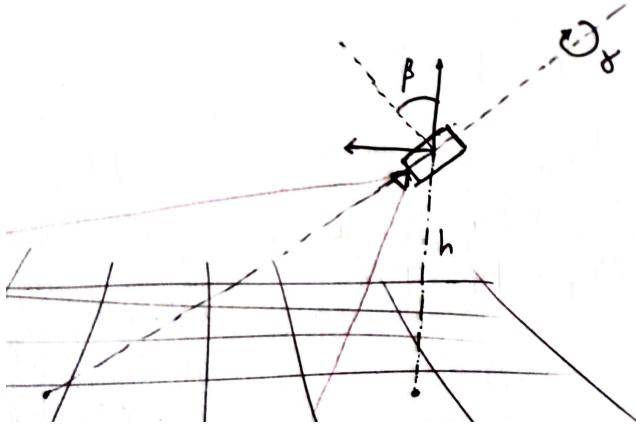


Figura 3: Esquema del problema cámara-plano planteado. Notar los tres parámetros extrínsecos que definen la posición y orientación de la cámara: altura, h , cabeceo, β , y $roll$, γ , respecto del plano.

3.1. Calculo de las ecuaciones paramétricas

Cada píxel de la fotografía refiere a una recta en el espacio 3D pudiendo estar los elementos fotografiados en cualquier punto contenido en cada recta de cada píxel. La determinación de las ecuaciones paramétricas de esas rectas es el primer objetivo, Figura 4.

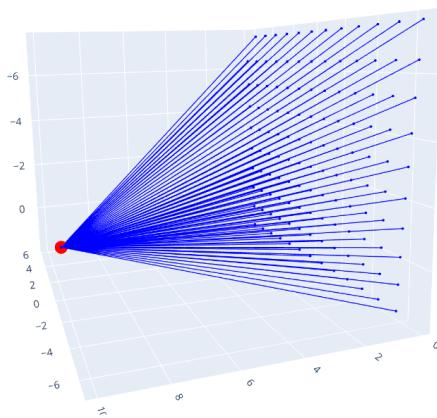


Figura 4: Rectas que salen de la cámara cuyas direcciones a cada píxel.

El hecho de que sean rectas permite que sus ecuaciones tengan un forma tal que,

$$\mathbf{r} = \begin{cases} x = x_0 + at, \\ y = y_0 + bt, \\ z = z_0 + ct. \end{cases} \quad (6)$$

donde t es el parámetro o grado de libertad que permite recorrer cierta recta.

Es posible simplificara tomando una de sus componentes como parámetro para recorrer la recta. Por comodidad

se toma, y , siendo esta la elección del eje de la cámara que coincidirá con su eje óptico. A su vez, podemos suponer que la cámara está en el origen del sistema de coordenadas y a una altura h sobre el plano tal que, $P = (0, 0, h)$, siendo su posición los términos independientes de la expresión de cada componente. Reescribiendo resulta,

$$\mathbf{r} = \begin{cases} x = pte_x \cdot y, \\ y = y, \\ z = pte_z \cdot y + h. \end{cases} \quad (7)$$

Ahora bien, se aprecian dos valores desconocidos, pte_x y pte_z , y la libertad del parámetro y . Este último será eliminado imponiendo la restricción de que los puntos deben estar contenidos en el plano más adelante. Sin embargo, las pendientes de las componentes x y z dependen intrínsecamente de la cámara (y de cada píxel de la imagen). Recorriendo la Ec. 5, la coordenada \mathbf{r} depende de la inversa de la matriz de la cámara linealmente siendo el factor de proporcionalidad particular para cada píxel la pendiente (pte_x y pte_z) necesaria en las ecuaciones paramétricas de las rectas, tal que,

$$\begin{aligned} \mathbf{r} &= M^{-1} \cdot r'_p \cdot y \rightarrow \\ \frac{d\mathbf{r}}{dy} &= M^{-1} \cdot r'_p = (pte_x, 1, pte_z) \end{aligned}$$

Se tiene que, como se advertía, las pendientes de las ecuaciones paramétricas son constantes, $pte_i(r'_p)$, es decir, rectas, siendo r'_p los índices desdistorsionados de un píxel de la imagen, tal que,

$$\begin{aligned} pte_x(r'_p) &= \frac{dx}{dy} = M_1^{-1} \cdot r'_p \\ pte_z(r'_p) &= \frac{dz}{dy} = M_3^{-1} \cdot r'_p \end{aligned}$$

Estas pendientes tienen un interpretación inmediata: son la tangente del ángulo de visión que abarca cada uno de los píxeles respecto al eje óptico. Siendo el FoV (Field of View) en un eje y otro (vertical y horizontal) el arco tangente de la pendiente de los píxeles extremos en un eje y otro. Esto es:

$$FoV_x = 2 \cdot \arctan(ppe_x(w_p, h_p/2)) \quad (8)$$

$$FoV_y = 2 \cdot \arctan(ppe_y(w_p/2, h_p)) \quad (9)$$

Donde w_p y h_p son la anchura y altura en píxeles de la imagen respectivamente. Algún ejemplo de cámaras calibradas se pueden apreciar en la Figura 10.

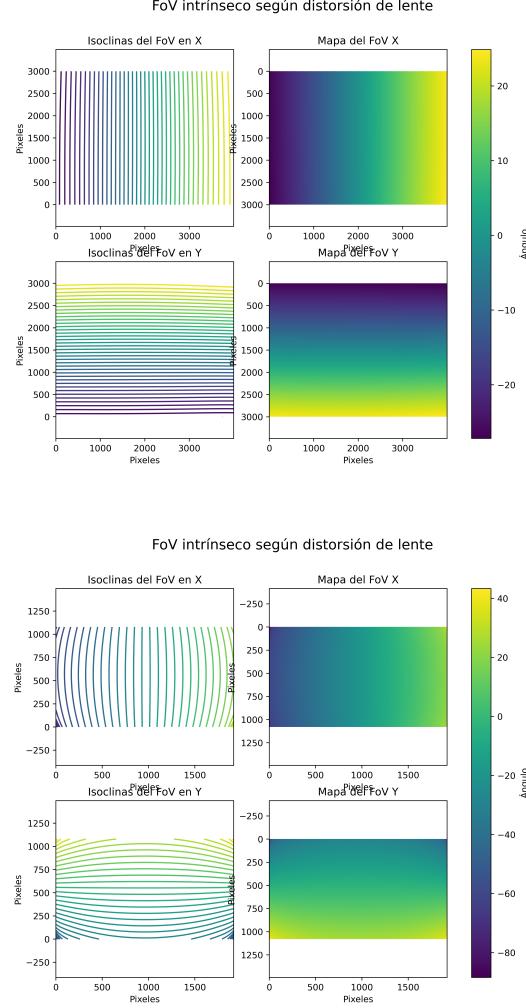


Figura 5: Ángulo que abarca cada píxel en el eje horizontal y vertical en la imagen respecto al eje óptico. Notar la deformación de lente de la segunda cámara.

3.2. Rotaciones de la cámara: cabeceo, β y roll, γ

Las ecuaciones de la forma Ec. 7 suponen un cámara con una posición extrínseca tal que el eje óptico este alineado con el eje y (paralelo al plano). Sin embargo, hay 3 grados de libertad en la orientación de un sólido rígido (una cámara) que podrían tenerse en cuenta. Uno de ello es prescindible por simplicidad: rotaciones en torno al eje normal al plano, en este caso eje z . Por ello, solo es necesario tener en cuenta la orientación extrínseca mediante dos ángulos: cabeceo, β , y roll, γ . Esto representan un giro respecto al eje x , el cual controla la incidencia del eje óptico al plano, la altura del horizonte en la imagen, la cenitalidad, etc. y el roll, γ , un giro respecto al eje y que controla el giro de la cámara sobre su propio eje, es decir, cómo de torcida se toma la

foto. Matemáticamente es posible aplicar una rotación respecto a un eje en 3D dimensiones mediante las matrices de rotación.

$$R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} \quad (10)$$

$$R_y(\gamma) = \begin{bmatrix} \cos(\gamma) & 0 & \sin(\gamma) \\ 0 & 1 & 0 \\ -\sin(\gamma) & 0 & \cos(\gamma) \end{bmatrix} \quad (11)$$

Tomando el vector posición, \mathbf{r} , definido mediante las ecuaciones paramétricas, Ec. 7 es posible aplicar las rotaciones una tras otra para “orientar” matemáticamente la cámara respecto al plano. Aplicando el giro entorno al eje óptico, es decir, el eje y , R_y ,

$$\begin{bmatrix} x_\gamma \\ y_\gamma \\ z_\gamma \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & 0 & \sin(\gamma) \\ 0 & 1 & 0 \\ -\sin(\gamma) & 0 & \cos(\gamma) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (12)$$

se tiene,

$$\mathbf{r}_\gamma = \begin{cases} x_\gamma = (pte_x \cos \gamma + pte_z \sin \gamma) \cdot y, \\ y_\gamma = y, \\ z_\gamma = (-pte_x \sin \gamma + pte_z \cos \gamma) \cdot y. \end{cases} \quad (13)$$

Posteriormente se aplica el giro en torno al eje x , provocando el cabeceo de la cámara,

$$\begin{bmatrix} x_{\gamma\beta} \\ y_{\gamma\beta} \\ z_{\gamma\beta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} x_\gamma \\ y_\gamma \\ z_\gamma \end{bmatrix} \quad (14)$$

se tiene,

$$\mathbf{r}_{\gamma\beta} = \begin{cases} x_{\gamma\beta} = (pte_x \cos \gamma + pte_z \sin \gamma) \cdot y, \\ y_{\gamma\beta} = [\cos \beta - \sin \beta (pte_z \cos \gamma - pte_x \sin \gamma)] \cdot y, \\ z_{\gamma\beta} = [\sin \beta + \cos \beta (pte_z \cos \gamma - pte_x \sin \gamma)] \cdot y. \end{cases} \quad (15)$$

3.3. Resolver las ecuaciones recta-plano

Una vez obtenidas las ecuaciones paramétricas al completo para cada píxel de la imagen es posible imponer la condición $z = 0$, es decir, que los puntos solución estén contenidos en el plano. Teniendo en cuenta un término independiente en la componente z debido a la altura, h , de la cámara sobre el plano y resolviendo el sistema de ecuaciones se llega a,

$$\mathbf{r} = \begin{cases} x = \frac{-h(pte_x \cos \gamma + pte_z \sin \gamma)}{\sin \beta + \cos \beta(pte_z \cos \gamma - pte_x \sin \gamma)} \\ y = \frac{-h(\cos \beta + \sin \beta(pte_x \sin \gamma - pte_z \cos \gamma))}{\sin \beta + \cos \beta(pte_z \cos \gamma - pte_x \sin \gamma)} \\ z = 0 \end{cases}, \quad (16)$$

Para visualizar los resultados para cada píxel de la imagen es posible representar los segmentos de varios de ellas desde el punto $(0, 0, h)$, es decir, la posición de la cámara hasta los puntos solución encontrados, Figura 6.

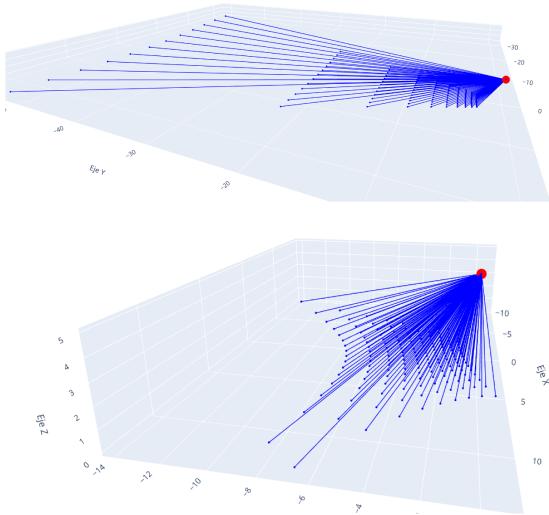


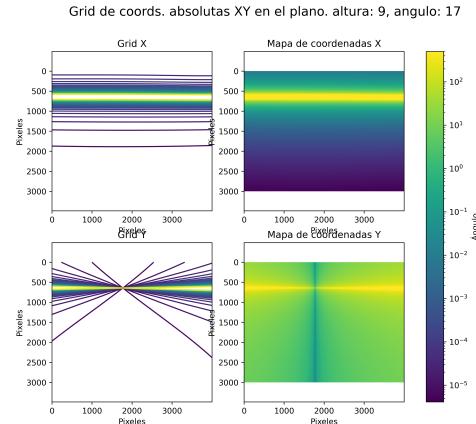
Figura 6: Solución de las ecuaciones paramétricas para una cámara con y sin deformación de lente para unas configuraciones extrínsecas dadas.

4. Interpretación y validación de resultados

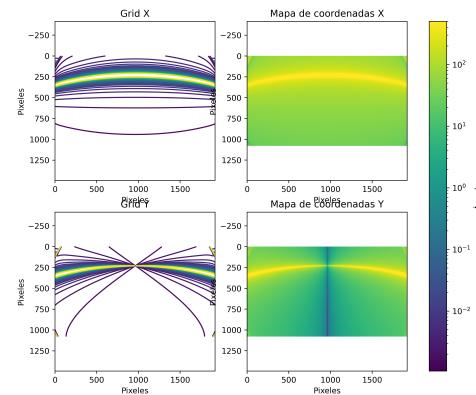
El cálculo realizado aspira a poder ubicar y medir elementos fotografiados que se encuentren contenidos en un plano de la realidad (un suelo, el mar, un solar, ...), por ello en este apartado se ofrecen ejemplos de uso validando su utilidad. Estos son el posicionar personas que andan por una calle gracias un detector de objetos basado en *deep learning* y medir el área de un folio.

4.1. Posicionamiento real de cada píxel

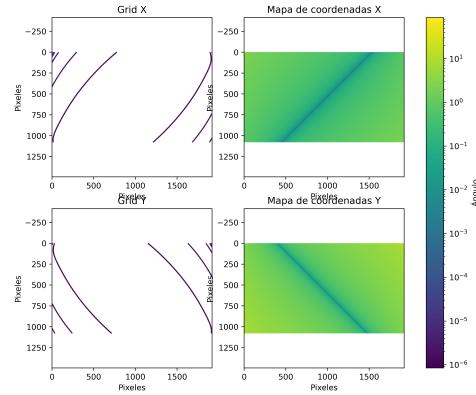
Una vez obtenidas las soluciones es posible representar el mapa de coordenadas (x, y) de la imagen para una configuración dada, Figura 7.



Grid de coords. absolutas XY en el plano. altura: 9, angulo: 17



Grid de coords. absolutas XY en el plano. altura: 9, angulo: 17



Grid de coords. absolutas XY en el plano. altura: 5.0, angulo: 90.0, roll: 45.0

Disponiendo de la correspondencia de cada píxel con una coordenada, $\mathbf{r} = (x, y)$, de la realidad sobre el plano es posible acoplarlo a un detector de objetos basado en *deep learning*. El modelo de IA proporciona el píxel en que se ubica en la imagen cierta instancia, y este no es otro que los índices, (i, j) , de la matriz de posiciones, \mathbf{r} , la cual propor-

ciona la posición real. Un ejemplo de uso se ha implementado con YOLOv8, [5] detectando personas. En la Figura 8 se aprecia sobre cada una sus coordenadas en metros respecto a la cámara.



Figura 8: Ubicación de las personas implementando YOLOv8 junto a la matriz de posiciones, r .

4.2. Cálculo de áreas reales

Una vez ubicados todos los píxeles de una imagen sobre el plano es posible medir distancias entre ellos o áreas. Para este último caso es posible calcular la tasa de cambio de posición o gradiente de un píxel a otro en los dos ejes de la imagen pudiendo obtener un Δx y Δy . De esta manera es posible calcular el área atribuida a cada uno tal que,

$$A_{ij} = \Delta x_{ij} \cdot \Delta y_{ij} \quad (17)$$

Esto resulta en un mapa de pesos, A_{ij} , con dimensiones $[L]^2[px]^{-2}$, es decir, una conversión área de píxeles a área real.

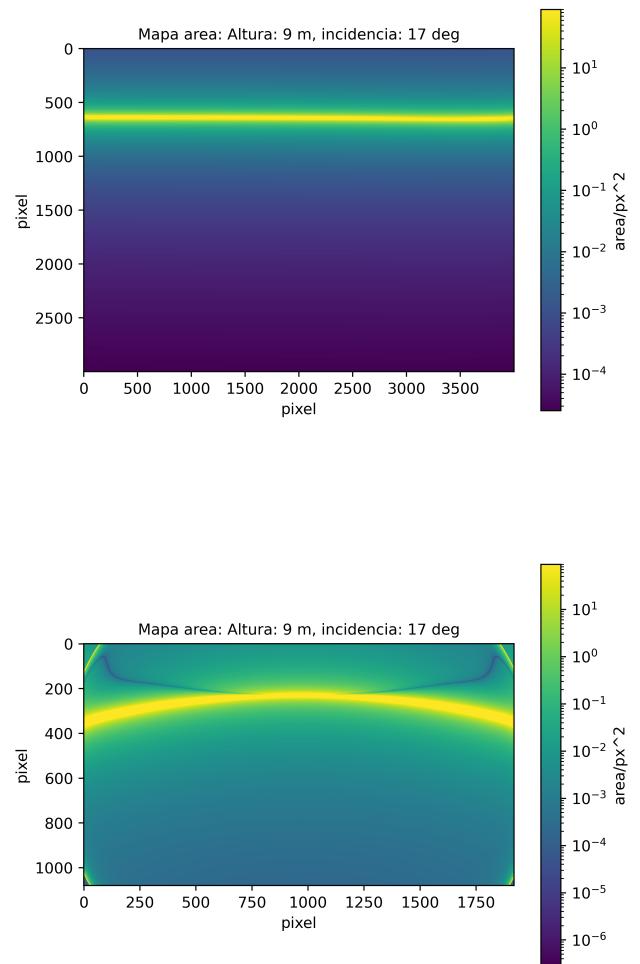


Figura 9: Mapa de pesos, A_{ij} , de conversión de áreas en píxeles a unidades reales, en este caso: m^2 para dos cámaras y disposiciones distintas.

Este mapa facilita en gran medida el cálculo de áreas, pudiendo multiplicar índice a índice de la matriz de pesos, A_{ij} , con una máscara binaria, M_{ij} , que perfila los elementos fotografiados, Ec. 18.

$$area = \sum_i \sum_j M_{ij} A_{ij} \quad (18)$$

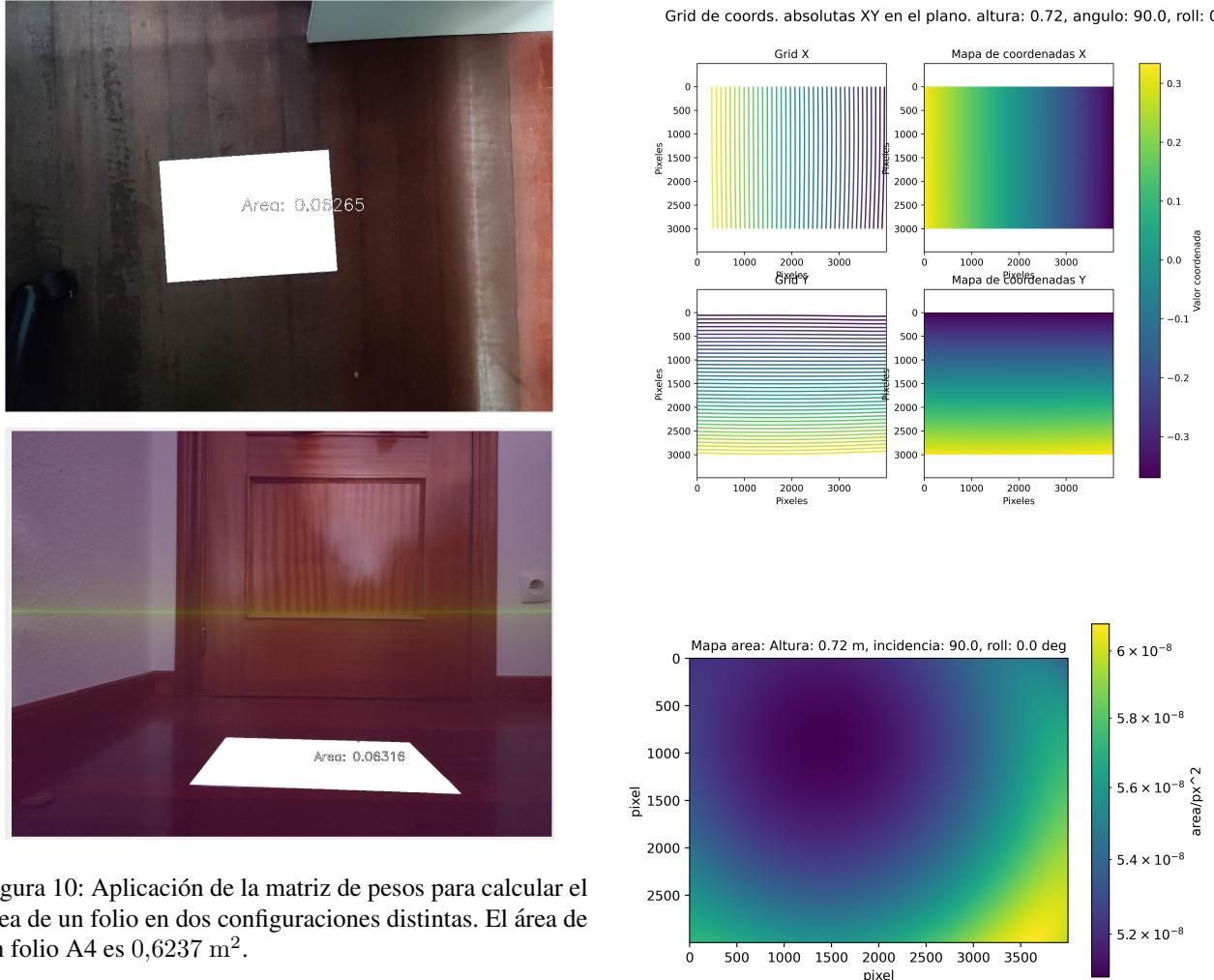


Figura 10: Aplicación de la matriz de pesos para calcular el área de un folio en dos configuraciones distintas. El área de un folio A4 es $0,6237 \text{ m}^2$.

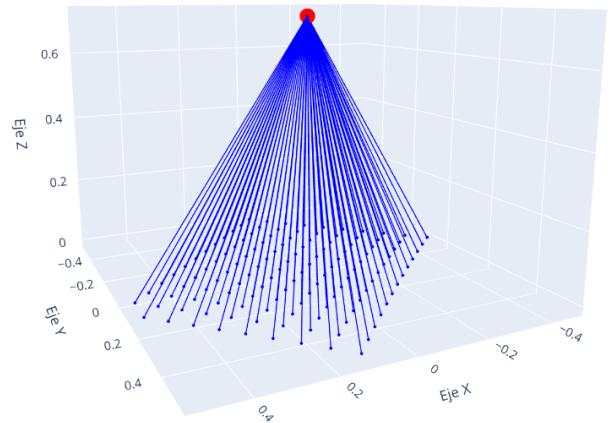


Figura 11: Solución para una cámara cenital sin deformación.

Para este último caso, en el que la foto es completamente cenital, esto es $\beta = 90^\circ$, se tendría los mapas de coordenadas y de pesos dados por la Figura 11.

Como es razonable, el eje x crece de izquierda a derecha y el y de arriba a abajo. El mapa de pesos que permite calcular el área tiene simetría radial, como cabía esperar.

5. Conclusión

Se ha obtenido un método para ubicar y medir elementos fotografiados con una cámara 2D sin conocer la profundidad a cada punto de la imagen. Es condición necesaria que los elementos estén contenidos en un plano y conocer la posición de la cámara respecto a este. Es válido para cualquier óptica. Se han ofrecido representaciones de los resultados, ejemplo de uso ubicando personas y una validación estimando el área de un folio A4 en dos configuraciones de cámara distintas.

Referencias

- [1] X.S. Gao, X.-R. Hou, J. Tang, H.-F. Chang Çomplete Solution Classification for the Perspective-Three-Point Problem”([96]). In this case the function requires exactly four object and image points.
- [2] F. Moreno-Noguer, V. Lepetit and P. Fua in the paper “^EPnP: Efficient Perspective-n-Point Camera Pose Estimation”
- [3] .^A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem”by G. Terzakis and M.Lourakis.
- [4] OpenCV, *Camera Calibration — OpenCV-Python Tutorials 1 documentation*, Disponible en: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.
- [5] Ultralytics, YOLO (*You Only Look Once*. Disponible en: <https://github.com/ultralytics>.