

Catálogo de objetos 3D para publicidad con realidad aumentada

^aÁngel Gabriel Rodríguez Rodríguez, Erwin Munguia Poblano, Rodrigo Olarte Astudillo

^bInstituto Politécnico Nacional, Av. Instituto Politécnico Nacional 1310, Tijuana, 22435, Baja California, México

Abstract

El presente proyecto tiene como objetivo la digitalización de objetos para usar los modelos obtenidos en un enfoque publicitario de realidad virtual. En este proyecto se aborda el problema de digitalizar productos, con la finalidad de que el usuario pueda acercar los productos a su vida diaria de manera accesible e incrementar la posibilidad de compra del producto. Para la digitalización de los objetos se emplea el método de proyección de franjas para la obtención de nubes de puntos 3D. Posteriormente, se emplearon las herramientas Blender, MeshLab y Unity para convertir las nubes de puntos en objetos con superficie sólida y así trasladarlo al entorno de realidad aumentada. Los resultados obtenidos confirman que este método es muy eficiente en cuanto tiempo pero aún requiere trabajo manual considerable. El diseño de métodos automáticos para fusionar nubes de puntos es un área de oportunidad para trabajo futuro.

Keywords: Visión por computadora, Realidad aumentada, Procesamiento de imágenes, Análisis de franjas, Desarrollo de fase

1. Introducción

1.1. Planteamiento del problema

La publicidad siempre ha sido un recurso importante que utilizan las empresas para acercar sus productos a los consumidores, y así lograr que estos se sintieran más cercanos con los suyos que con los de la competencia.

La industria publicitaria no se ha quedado constante a lo largo de los años, sino que, al igual que la tecnología, ha estado en constante evolución para poder adaptarse a las necesidades actuales de los consumidores. Sin embargo, cuando existen muchas empresas que desean acercar sus productos a los consumidores, estas deben buscar nuevas medidas que logren que sus productos sobresalgan por encima de los competidores.

La problemática que se aborda en este proyecto es encontrar una medida para llevar una real aproximación del producto a los consumidores mediante modelos 3D de los productos y realidad aumentada.

1.2. Objetivos

El objetivo general es implementar una aplicación móvil que permita mostrar objetos en realidad aumentada de productos usando la técnica de proyección estructurada de franjas.

Los objetivos específicos de este proyecto se enumeran a continuación:

1. Implementar la técnica proyección estructurada de franjas.
2. Realizar una aplicación móvil de android que permita visualizar los productos en realidad aumentada.
3. Digitalizar productos para obtener su modelo 3D.

2. Marco teórico

2.1. Coordenadas homogéneas

Para entender como función la digitalización 3D mediante la proyección digital de franjas es necesario primero entender que son las coordenadas homogéneas, pero ¿Qué son?, En la referencia (1) se explica que cuando se representa un punto en un espacio n-dimensional éste se define por un vector de la forma

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T, \quad (1)$$

donde $[\cdot]^T$ denota la transpuesta. Entonces para obtener las coordenadas homogéneas se añade una entrada extra a “ \mathbf{x} ” con un valor equivalente a la unidad. Este proceso genera un vector de $(n + 1)$ dimensiones, y puede representarse por el operador \mathcal{H} definido como

$$\mathcal{H}[\mathbf{x}] = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \quad (2)$$

Así mismo en las coordenadas homogéneas la última entrada del vector es conocida como la escala. Es conveniente definir otro operador con el cual se pueda obtener la última entrada de cualquier vector dado.

$$x_n = S[\mathbf{x}], \quad y \quad 1 = S[\mathcal{H}[\mathbf{x}]]. \quad (3)$$

Adicionalmente, cabe notar que el operador \mathcal{H} coloca la escala a la unidad, debido a ello se necesita otro operador que coloque la escala a 0, de modo que, se define el operador

$$\mathcal{H}_0[\mathbf{x}] = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}. \quad (4)$$

Cabe notar que el operador \mathcal{H}_0 no afecta la dirección ni la norma de \mathbf{x} . Los operadores \mathcal{H} y \mathcal{H}_0 pueden ser considerados como

dos casos particulares de un operador general definido como

$$\mathcal{H}_s[\mathbf{x}] = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix}, \quad (5)$$

donde s es cualquier escalar. Para revertir el proceso de añadir una entrada extra al vector, solo es necesario retornar el vector dado a excepción de su última entrada, para esto se define el operador inverso \mathcal{H}_0^{-1} como se muestra a continuación. Para cualquier vector $(n + 1)$ -dimensional

$$\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_n \ y_{n+1}]^T, \quad (6)$$

el operador \mathcal{H}_0^{-1} se define entonces

$$\mathcal{H}_0^{-1}[\mathbf{y}] = [y_1 \ y_2 \ \cdots \ y_n]^T. \quad (7)$$

Basándose entonces en el operador \mathcal{H}_0^{-1} , el operador inverso para \mathcal{H}_s con $s \neq 0$ es definido como

$$\mathcal{H}_s^{-1}[\mathbf{y}] = \frac{s}{S[\mathbf{y}]} \mathcal{H}_0^{-1}[\mathbf{y}], \quad (8)$$

Infiriendo entonces que el operador \mathcal{H}^{-1} será referido como el operador homogéneo inverso. La inversa \mathcal{H}_0^{-1} es un operador lineal. Esto significa que para cualesquiera dos escalares λ_1 y λ_2 , tenemos que

$$\mathcal{H}_0^{-1}[\lambda_1 \mathbf{y}_1 + \lambda_2 \mathbf{y}_2] = \lambda_1 \mathcal{H}_0^{-1}[\mathbf{y}_1] + \lambda_2 \mathcal{H}_0^{-1}[\mathbf{y}_2]. \quad (9)$$

Por otro lado, el operador \mathcal{H}_s^{-1} es invariante a multiplicaciones escalares no cero a su argumento.

$$\mathcal{H}_s^{-1}[\lambda \mathbf{y}] = \mathcal{H}_s^{-1}[\mathbf{y}], \quad s, \lambda \neq 0. \quad (10)$$

Los operadores \mathcal{H}_s y \mathcal{H}_s^{-1} pueden ser expresados en términos del operador homogéneo y su inversa, concretamente

$$\mathcal{H}_s = \Xi_s \mathcal{H}[\mathbf{x}] \quad (11)$$

$$\mathcal{H}_s^{-1}[\mathbf{y}] = \mathcal{H}^{-1}[\Xi_s^{-1} \mathbf{y}], \quad s \neq 0, \quad (12)$$

donde

$$\Xi_s = \begin{bmatrix} \mathbb{I}_n & \mathbf{0}_n \\ \mathbf{0}_n^T & s \end{bmatrix}, \quad (13)$$

y \mathbb{I}_n es la matriz de identidad $n \times n$.

2.2. Modelo pinhole

Entendiendo como funcionan las coordenadas homogéneas ahora procedemos a explicar el modelo pinhole. En la práctica, la captura de imágenes es realizado por un dispositivo equipado con una lente, ya que este dispositivo produce imágenes de alta calidad debido al sistema de lentes que minimiza la aberración y la distorsión, sin embargo el procesamiento de imágenes puede ser modelado usando únicamente una lente delgada, además, el modelado de imágenes puede derivarse fácilmente utilizando su equivalente, la cámara estenopeica.

En una cámara estenopeica, el origen del sistema de coordenadas esta fijado en la apertura de la cámara y el eje z es

paralelo al eje óptico, entonces el plano $z = -f$, donde f es la distancia focal, en el plano de la imagen actual. Cabe notar que la imagen se invierte, por lo tanto, los ejes de las ordenadas y abscisas son revertidos para describir la imagen como la versión magnificada del objeto. Para evitar la inversión de los ejes se usa el plano conjugado de la imagen en la posición $z = f$

En una cámara estenopeica centrada, el sistema de coordenadas $O_c x_c y_c z_c$ es conocido como el marco de referencia de la cámara. Desde este sistema de referencia, el punto observado está dado por el vector de coordenadas

$$\mathbf{p}_c = [x_c \ y_c \ z_c]^T \quad (14)$$

El punto \mathbf{p}_c será detectado en el plano de la imagen $z_c = f$ en el punto

$$\boldsymbol{\beta} = [\beta_x \ \beta_y]^T, \quad (15)$$

donde $\boldsymbol{\beta}$ será referido como las coordenadas físicas de la imagen. Siendo así que el modelo de proyección pinhole relaciona los vectores \mathbf{p}_c y $\boldsymbol{\beta}$ mediante

$$\begin{bmatrix} \beta_x \\ \beta_y \\ f \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}. \quad (16)$$

Usando coordenadas homogéneas para el punto de la imagen $\boldsymbol{\beta}$, puede ser reescrito como

$$\boldsymbol{\beta} = \mathcal{H}_f^{-1}[\mathbf{p}_c] \quad (17)$$

$$= \mathcal{H}^{-1}[\Xi_f^{-1} \mathbf{p}_c]. \quad (18)$$

La imagen formada en el sensor de la cámara es muestreada como un arreglo de píxeles. Entonces, las coordenadas físicas $\boldsymbol{\beta}$ serán transformadas a las coordenadas píxel

$$\boldsymbol{\mu} = [u \ v]^T, \quad (19)$$

las cuales dependen del tamaño del píxel y la desviación (distorsión diagonal). El muestreo puede ser descrito como

$$u = (\beta_x + \tau_x)/s_x + \sigma \beta_y, \quad (20)$$

$$v = (\beta_y + \tau_y)/s_y, \quad (21)$$

donde s_x y s_y (con unidades de longitud, por ejemplo, milímetros) son el ancho y la altura del píxel, respectivamente, $\boldsymbol{\tau} = [\tau_x, \tau_y]^T$ es conocido como el punto principal y representa el punto (desde el marco de referencia uv) donde el eje óptico atraviesa el plano de la imagen, y σ es el factor de desviación ($\sigma = 0$ para la mayoría de los sensores de las cámaras), entonces las ecuaciones(20) y (21) pueden ser escritas como

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & \sigma & \tau_x/s_x \\ 0 & 1/s_y & \tau_y/s_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_x \\ \beta_y \\ 1 \end{bmatrix}, \quad (22)$$

o usando una notación compacta

$$\boldsymbol{\mu} = \mathcal{H}^{-1}[S\mathcal{H}[\boldsymbol{\beta}]] = \mathcal{P}_s[\boldsymbol{\beta}], \quad (23)$$

donde S es la matriz de muestreo dada como

$$S = \begin{bmatrix} 1/s_x & \sigma & \tau_x/s_x \\ 0 & 1/s_y & \tau_y/s_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (24)$$

sustituyendo la ecuación (18) en la ecuación (23), obtenemos la imagen $\boldsymbol{\mu}$ (en coordenadas píxel) del punto \mathbf{p}_c como

$$\boldsymbol{\mu} = \mathcal{P}_s[\mathcal{H}^{-1}[\Xi_f^{-1}\mathbf{p}_c]] \quad (25)$$

$$= \mathcal{H}^{-1}[S\Xi_f^{-1}\mathbf{p}_c] \quad (26)$$

$$= \mathcal{H}^{-1}[K\mathbf{p}_c], \quad (27)$$

donde $K = S\Xi_f^{-1}$ conocida como la matriz de parámetros extrínsecos teniendo la forma explícita

$$K = \begin{bmatrix} 1/s_x & \sigma & \tau_x/f s_x \\ 0 & 1/s_y & \tau_y/f s_y \\ 0 & 0 & 1/f \end{bmatrix}. \quad (28)$$

Por el contrario, si consideramos que la cámara estenopeica esta en una posición y orientación arbitraria con respecto al sistema de coordenadas de referencia. La posición y orientación de la cámara son definidas por el vector \mathbf{t} y la rotación de la matriz R , respectivamente.

Considerando que $\mathbf{p} = [x \ y \ z]^T$ es un punto en el sistema de coordenadas de referencia global, entonces el punto \mathbf{p} es visto desde el sistema de referencia de la cámara como

$$\mathbf{p}_c = \mathcal{R}^T(\mathbf{p} - \mathbf{t}) = L\mathcal{H}[\mathbf{p}], \quad (29)$$

donde L es conocido como la matriz de parámetros extrínsecos teniendo la forma explicita

$$L = [R^T \quad -R^T\mathbf{t}]. \quad (30)$$

Sustituyendo la ecuación (29) en la ecuación (25), el proceso de representación completo por una cámara estenopeica no centrada es dado por

$$\boldsymbol{\mu} = \mathcal{H}^{-1}[KL\mathcal{H}[\mathbf{p}]] \quad (31)$$

$$= \mathcal{H}^{-1}[C\mathcal{H}[\mathbf{p}]], \quad (32)$$

donde $C = KL$ es la matriz de la cámara.

De manera general la ecuación (32) describe la transformación de puntos \mathbf{p} de un espacio tridimensional a un espacio bidimensional.

En el caso particular donde \mathbf{p} representa los puntos de un plano en un espacio tridimensional; podemos establecer una parametrización bidimensional como

$$\mathbf{p} = \Pi\mathcal{H}[\boldsymbol{\rho}], \quad (33)$$

donde $\boldsymbol{\rho} = [\rho_x, \rho_y]^T$ parametriza el plano, $\Pi = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{s}]$ es la matriz del plano, \mathbf{q}_1 y \mathbf{q}_2 son columnas de la matriz de rotación

$Q = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]^T$, y \mathbf{s} es un vector de translación. Continuando, los puntos \mathbf{p} son transformados a $\boldsymbol{\mu}$ por la ecuación (32) como

$$\boldsymbol{\mu} = \mathcal{H}^{-1}[C\mathcal{H}[\Pi\mathcal{H}[\boldsymbol{\rho}]]] \quad (34)$$

$$= \mathcal{H}^{-1}\left[C\begin{bmatrix} \Pi \\ \mathcal{H}[\mathbf{0}_2]^T \end{bmatrix}H[\boldsymbol{\rho}]\right] \quad (35)$$

$$= \mathcal{H}^{-1}[G\mathcal{H}[\boldsymbol{\rho}]] \quad (36)$$

$$= \mathcal{P}_G[\boldsymbol{\rho}], \quad (37)$$

donde G es conocida como la matriz homografía y tiene la forma explícita

$$G = KL\begin{bmatrix} \Pi \\ \mathcal{H}[\mathbf{0}_2]^T \end{bmatrix} \quad (38)$$

$$= KR^T[\Pi \quad -\mathbf{t}\mathcal{H}[\mathbf{0}]^T] \quad (39)$$

$$= KR^T\bar{\Pi}, \quad (40)$$

donde

$$\bar{\Pi} = [\Pi \quad -\mathbf{t}\mathcal{H}[\mathbf{0}]^T] \quad (41)$$

$$= [\mathbf{q}_1 \quad \mathbf{q}_1 \quad \mathbf{s} - \mathbf{t}]. \quad (42)$$

2.3. Demodulación de fase

El sistema de proyección de franjas es uno de los métodos más empleados para la digitalización de objetos, debido a las características que ofrece, como su rápido procesamiento, su alta exactitud y su operación simple. Así mismo, esta técnica cuenta con un proceso llamado “Demodulación de fase” que está formado por tres pasos.

1. Preprocesamiento
2. Extracción de fase envuelta
3. Desenvolvimiento de fase

Como resultado obtendremos los puntos del plano del objeto a su posición correspondiente en el plano de la superficie.

2.3.1. Generación de las franjas

Durante el preprocesamiento se tienen que realizar diversas tareas para obtener los mejores resultados en los pasos posteriores. Algunas de estas actividades que se deben realizar es la limpieza del ruido, la definición de la región objetivo y la proyección del patrón de franjas.

De forma general, el patrón de franjas es desplegado por el proyector sobre el objeto que se desea digitalizar, esto con la finalidad de que la cámara capte la distorsión de las franjas sobre el objeto. Para la creación de dichas franjas puede ser descrita de la forma (2)

$$I(x, y) = \alpha(x, y) + \beta(x, y)\cos[\phi(x, y) + \delta] + \eta(x, y), \quad (43)$$

En donde, la coordenada (x, y) es un punto cualquiera del patrón de franjas, α representa la luz del fondo, β la luz de modulación, ϕ la fase codificada, δ representa una fase adicional, que comúnmente se le conoce como cambio de fase y η es un ruido aleatorio que puede ser opcional al momento de representar la ecuación.

2.3.2. Extracción de la fase usando mínimos cuadrados

Una vez que se haya realizado la captura de patrones de franjas sobre la superficie del objeto, el siguiente paso consta en extraer la fase envuelta, en este paso se recibe (3) una o más franjas que fueron capturadas anteriormente y están descritos por la ecuación 43

Para la extracción de la fase envuelta existen múltiples funciones llamados como “Algoritmos de cambio de fase”, estos tienen como propósito obtener $\alpha(x, y)$, $\beta(x, y)$ y $\phi(x, y)$ del patrón de franjas.

Algunos de los algoritmos que se utilizan dependen directamente del número de cambios que tuvo el patrón de franjas sobre el objeto, por ejemplo, el “algoritmo de tres pasos” se usa cuando el patrón de franjas cuenta con 3 corrimientos de fase de la forma (2)

$$\delta_k = \frac{2\pi}{3}(k-1), \quad k = 1, 2, 3. \quad (44)$$

El “algoritmo de cuatro pasos”, es de utilidad cuando cuando existen cuatro corrimientos de fase en el patrón que tengan la forma de (2)

$$\delta_k = \frac{\pi}{2}(k-1), \quad k = 1, 2, 3, 4. \quad (45)$$

Existen otros algoritmos que sirven para cualquier número de corrimientos de fase del sistema, como el “algoritmo de n pasos”, (2) siempre y cuando el patrón de franjas esté dado uniformemente en $[0, 2\pi p]$ de la manera siguiente

$$\delta_k = p \frac{2\pi}{n}(k-1), \quad k = 1, 2, 3, \dots, n. \quad (46)$$

Sin embargo, a pesar de todos estos algoritmos, el que se usará para este proyecto será el “algoritmo de mínimos cuadrados”, Según se menciona en (2), los algoritmos previamente vistos únicamente sirven cuando los corrimientos de fase cuentan con una función lineal, sin embargo, no siempre esta condición se puede cumplir al momento del envolvimiento de fase, por lo que se necesita de otra función que pueda ser usada sin importar que función se usa para las fases del patrón de franjas, y uno de estos algoritmos que cumple con la condición es el algoritmo de mínimos cuadrados.

Esta función se usa cuando el el corrimiento de fase puede ser escrita de la siguiente forma (2)

$$Y = AC, \quad (47)$$

donde

$$Y = \begin{bmatrix} I_1(x_1) & I_1(x_2) & \dots & I_1(x_N) \\ I_2(x_1) & I_2(x_2) & \dots & I_2(x_N) \\ \vdots & \vdots & \dots & \vdots \\ I_n(x_1) & I_n(x_2) & \dots & I_n(x_N) \end{bmatrix}, \quad (48)$$

$$A = \begin{bmatrix} 1 & \cos\delta_0 & -\sin\delta_0 \\ 1 & \cos\delta_1 & -\sin\delta_1 \\ \vdots & \vdots & \vdots \\ 1 & \cos\delta_n & -\sin\delta_n \end{bmatrix}, \quad (49)$$

$$C = \begin{bmatrix} a(x_1) & \dots & a(x_N) \\ b(x_1)\cos\phi(x_1) & \dots & b(x_N)\cos\phi(x_N) \\ b(x_1)\sin\phi(x_1) & \dots & b(x_N)\sin\phi(x_N) \end{bmatrix}, \quad (50)$$

A su vez, se puede representar a C como

$$C = \begin{bmatrix} c_1(x) \\ c_2(x) \\ c_3(x) \end{bmatrix}, \quad (51)$$

$$C = (A^T A)^{-1} A^T Y. \quad (52)$$

Una vez teniendo esto, podemos obtener el resultado del envolvimiento de fase, que es obtener los valores de α , β y ϕ , siendo estos (2)

$$\alpha(x) = c_1(x), \quad (53)$$

$$\beta(x) = \sqrt{c_2^2(x) + c_3^2(x)}, \quad (54)$$

$$\phi(X) = \arctan\left(\frac{c_3(X)}{c_2(X)}\right). \quad (55)$$

2.3.3. Desenvolvimiento de fase

Por último, una vez que hayamos realizado el envolvimiento de fase, se menciona en (3) que se debe realizar un último paso para eliminar todas las posibles discontinuidades que estén presentes en nuestro resultado obtenido por el paso anterior, a esto se le conoce como el desenvolvimiento de fase, el cual se representa de la manera (3) (4)

$$\phi_k = \psi_k + 2\pi\tilde{k}, \quad (56)$$

$$\tilde{k} = \text{round}\left(\frac{\alpha_{k-1}\phi_{k-1}(\mu) - \psi_k(\mu)}{2\pi}\right), \quad (57)$$

Donde ψ es la fase envuelta, ϕ es el resultado del desenvolvimiento de fase y \tilde{k} es la función de aproximación.

Todo este procedimiento puede ser repetido un número k de veces, por lo tanto, se puede apreciar esto como (4)

$$f_k = \alpha_{k-1} f_{k-1}, \quad (58)$$

Entonces, para obtener ϕ_k se necesita de ϕ_{k-1} siempre que $k > 1$, en caso contrario, si se trata de la primera fase, solamente se usa

$$\phi = \psi. \quad (59)$$

2.4. Calibración

Ya que hemos obtenido la fase desenvuelta del objeto a digitalizar, debemos conocer dónde se encuentra físicamente en el plano, esto lo logramos con la técnica de triangulación empleado la cámara y el proyector. Sin embargo, antes de poder efectuar una triangulación, debemos conocer los parámetros intrínsecos y extrínsecos de la cámara y del proyector que estemos usando.

El método de calibración empleado en este proyecto está basado en el modelo pinhole descrito en (5) el cual refiere a que

teniendo un punto en el espacio 3D $\rho = [x, y, x]^T$, podemos obtener su correspondiente μ en la imagen mediante la ecuación:

$$\mu = \mathcal{H}^{-1}[D\mathcal{H}[\rho]] \quad (60)$$

donde \mathcal{H} es el operador de coordenadas homogéneas y D es:

$$D = \lambda K[R^T, -R^T t]. \quad (61)$$

Nuestra labor es encontrar la matriz K de parámetros intrínsecos de tamaño 3×3 , la matriz de rotación R de 3×3 y el vector de traslación desde el origen de nuestro plano de referencia t , para lograrlo usaremos homografías (5), la cual se denota:

$$G = \lambda K[\bar{r}_1, \bar{r}_2, -R^T t], \quad (62)$$

dicha homografía puede ser calculada teniendo los puntos de la imagen y los puntos originales en nuestro plano de referencia, se puede también describir como una matriz de 3×3 :

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}, \quad (63)$$

podemos obtener la homografía mediante la ecuación

$$Ag = 0_{2m}, \quad (64)$$

donde g son todas las incógnitas de la homografía de forma $g = [g_{11}, g_{12}, g_{13}, g_{21}, g_{22}, g_{23}, g_{31}, g_{32}, g_{33}]^T$.

Denotamos a la matriz A como la matriz de regresión y se construye con los puntos ρ del patrón de referencia y los puntos μ de la imagen capturada, de la siguiente manera:

$$A = \begin{bmatrix} \mathcal{H}[\rho_1]^T & 0_3^T & -\mu_{x1}\mathcal{H}[\rho_1]^T \\ \vdots & \vdots & \vdots \\ \mathcal{H}[\rho_m]^T & 0_3^T & -\mu_{xm}\mathcal{H}[\rho_m]^T \\ 0_3^T & \mathcal{H}[\rho_1]^T & -\mu_{y1}\mathcal{H}[\rho_1]^T \\ \vdots & \vdots & \vdots \\ 0_3^T & \mathcal{H}[\rho_m]^T & -\mu_{ym}\mathcal{H}[\rho_m]^T \end{bmatrix}. \quad (65)$$

Para evitar la solución trivial del sistema, la solución planeada en (5) es mediante la descomposición en valores singulares (SVD), si $A = U_A S_A V_A$, entonces la solución de mínimos cuadrados para g es la última columna de V_A .

2.4.1. Obtención de parámetros intrínsecos

De acuerdo con (5), la cantidad mínima de homografías para calcular la matriz K debe ser 4, nuestras homografías son de la forma:

$$G_k = \lambda_k K[\bar{r}_{1,k}, \bar{r}_{2,k}, -R_k^T t_k], \quad k \in [1, \dots, n] \quad (66)$$

para facilitar el cálculo, usaremos las dos primeras columnas de nuestras homografías, dichas columnas no contienen al vector t de traslación, denotamos estas columnas como:

$$\begin{aligned} g_{1,k} &= \lambda_k K \bar{r}_{1,k}, \\ g_{2,k} &= \lambda_k K \bar{r}_{2,k}. \end{aligned} \quad (67)$$

Debido a la ortogonalidad de la matriz de rotación podemos aplicar las propiedades $\bar{r}_{1,k}^T \bar{r}_{2,k} = 0$, $\bar{r}_{1,k}^T \bar{r}_{1,k} = \bar{r}_{2,k}^T \bar{r}_{2,k}$, teniendo:

$$\begin{aligned} g_{1,k}^T W g_{2,k} &= 0, \\ g_{1,k}^T W g_{1,k} - g_{2,k}^T W g_{2,k} &= 0, \end{aligned} \quad (68)$$

donde $W = K^{-T} K^{-1}$ conocida como imagen de la cónica absoluta (5). Debido a que W es simétrica y de tamaño 3×3 , solo tiene 6 incógnitas por encontrar,

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_2 & w_4 & w_5 \\ w_3 & w_5 & w_6 \end{bmatrix} \quad (69)$$

Para calcular la solución $w = [w_1, w_2, w_3, w_4, w_5, w_6]$ debemos usar la ecuación $Bw = 0_{2n}$, donde,

$$B = \begin{bmatrix} v_{1,2}^T[G_1] \\ v_{1,2}^T[G_2] \\ \vdots \\ v_{1,2}^T[G_n] \\ v_{1,1}^T[G_1] - v_{2,2}^T[G_1] \\ v_{1,1}^T[G_2] - v_{2,2}^T[G_2] \\ \vdots \\ v_{1,1}^T[G_n] - v_{2,2}^T[G_n] \end{bmatrix}, \quad (70)$$

siendo v :

$$v_{ij}[G] = \begin{bmatrix} g_{1i}g_{1j} \\ g_{2i}g_{2j} \\ g_{3i}g_{3j} \\ g_{2i}g_{1j} + g_{1i}g_{2j} \\ g_{3i}g_{1j} + g_{1i}g_{3j} \\ g_{3i}g_{2j} + g_{2i}g_{3j} \end{bmatrix}. \quad (71)$$

La solución a este sistema es obtenida empleando SVD, calculando el vector w para posteriormente obtener la matriz W , a partir de esta matriz podemos calcular K usando la descomposición de Cholesky (5).

2.4.2. Obtención de parámetros extrínsecos

Una vez que hemos calculado la matriz K entonces ya solo debemos calcular la matriz de rotación R y el vector de traslación t . De acuerdo con (5) estos parámetros son obtenidos mediante

$$H_k = K^{-1} G_k, \quad (72)$$

la cual también expresamos como columnas

$$[b_{1,k}, b_{2,k}, b_{3,k}] = [\lambda_k \bar{r}_{1,k}, \lambda_k \bar{r}_{2,k}, -\lambda_k R_k^T t_k]. \quad (73)$$

Una estimación de nuestra matriz de rotación es:

$$\begin{aligned} \bar{R}_k^T &= [b_{1,k}, b_{2,k}, b_{1,k} \times b_{2,k}] \\ &= [\lambda_k \bar{r}_{1,k}, \lambda_k \bar{r}_{2,k}, \lambda_k^2 \bar{r}_{3,k}]. \end{aligned} \quad (74)$$

Usando el método SVD obtenemos la matriz de rotación, $\bar{R}_k = U_R S_R V_R^T$, y de acuerdo con (5), la matriz de rotación termina siendo:

$$R_k = U_R V_R^T. \quad (75)$$

Cuando hemos calculado la matriz de rotación, debemos estimar los escalares λ para cada homografía, de la siguiente manera:

$$\lambda_k = (r_{1,k}^{-T} b_{1,k} + r_{2,k}^{-T} b_{2,k})/2, \quad (76)$$

finalmente obtenemos el vector de traslación con la siguiente ecuación (5):

$$t_k = -R_k b_{3,k} / \lambda_k. \quad (77)$$

2.4.3. Obtención de puntos del patrón

Para obtener las homografías útiles al momento de calibrar es necesario contar con puntos ρ en el plano de referencia (el patrón) y puntos μ en la imagen capturada (píxeles), como mínimo se necesitan cuatros pares de puntos de correspondencia. (6).

El patrón empleado es de tipo ajedrez como se muestra en la Figura 1 (a), para calibrar la cámara y el proyector requerimos tomar diversas fotos a este patrón como se ha mencionado, sin embargo tenemos el problema de que el proyector no es capaz de tomar fotos a diferencia de la cámara.

Entonces para calibrar el sistema se emplea el método descrito en (6), donde se emplean dos patrones, uno impreso de color amarillo y el otro proyectado de color cian. De esta forma, se pueden recuperar los dos patrones de calibración individualmente al separar los canales de color de las imágenes resultantes, como se ve en la Figura 1.

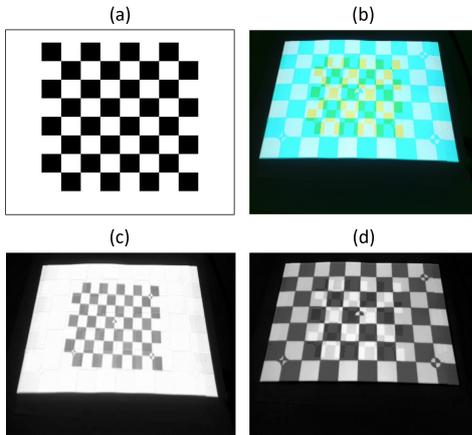


Figura 1: (a) Patrón de tipo ajedrez para calibración. (b) patrón de color amarillo impreso sobre una superficie y patrón de color cian proyectado superpuesto. (c) Componente de color azul en escala de grises. (d) Componente de color rojo en escala de grises.

2.4.4. Calibración de la cámara

Requerimos los puntos del patrón de calibración, tanto en el plano de referencia, ρ en milímetros, como en la imagen, μ en píxeles, del patrón descrito en la Figura 1 (c) se detectan todos los puntos internos, 54 puntos, tanto de la captura por la cámara como calculadas en el plano de referencia conociendo el tamaño de los cuadrados.

2.4.5. Calibración del proyector

La calibración del proyector es más complicada debido a que este no puede capturar imágenes como lo hace cámara, es por ello que mediante el método descrito en (5) se proyecta el patrón y capturamos imágenes con la cámara, los puntos μ son obtenidos mediante aritmética de la imagen que se proyecta conociendo el tamaño de los cuadros. Los puntos ρ son calculados mediante de la siguiente manera:

$$\rho = \mathcal{H}^{-1}[G_a^{-1}\mathcal{H}[\mu]], \quad (78)$$

donde G_a es la homografía que obtenemos usando los puntos μ de la captura del patrón con la cámara.

2.5. Triangulación

Finalmente tenemos el proceso de la triangulación; según (7) la proyección de dos ejes emplea las franjas μ y ν proyectando líneas l_μ y l_ν para iluminar el objeto a medir. Este enfoque permite obtener las coordenadas (μ, ν) de puntos s en la superficie de la diapositiva del proyector. Por lo tanto, un punto p observado por el sistema se estima por intersección lineal. Obsérvese que la proyección en dos ejes aporta la siguiente redundancia conveniente para estimar los puntos observados por el sistema FPP. Cada dispositivo proporciona dos restricciones a través del píxel de coordenadas $s = [\mu, \nu]^T$, mientras que el vector $p = [x, y, z]^T$ que debe estimarse sólo tiene tres incógnitas. En otras palabras, los sistemas FPP con m cámaras y n proyectores producen $2(m+n)$ restricciones para estimar tres incógnitas. Esta redundancia es conveniente porque las coordenadas s de los proyectores se obtienen mediante demodulación de fase, que se ve afectada por el ruido y otras fuentes de error experimental.

Las principales ventajas de la proyección en dos ejes son la robustez y la simplicidad de los algoritmos de intersección de líneas para la estimación de puntos tridimensionales. Sin embargo, la proyección de ambas franjas consume mucho tiempo y es excesivo para algunas aplicaciones.

El proceso para obtener esta triangulación lineal es mediante la resolución de un sistema de ecuaciones descrito de la siguiente forma

$$p = t_1 + \lambda_1 R_1 K_1^{-1} \mathcal{H}[\mu], \quad (79)$$

$$p = t_2 + \lambda_2 R_2 K_2^{-1} \mathcal{H}[\nu], \quad (80)$$

el cual puede ser reescrito de la siguiente manera en una matriz de regresión

$$t_1 + \lambda_1 R_1 K_1^{-1} \mathcal{H}[\mu] = t_2 + \lambda_2 R_2 K_2^{-1} \mathcal{H}[\nu] \quad (81)$$

$$\lambda_1 R_1 K_1^{-1} \mathcal{H}[\mu] - \lambda_2 R_2 K_2^{-1} \mathcal{H}[\nu] = t_2 - t_1 \quad (82)$$

$$\begin{bmatrix} R_1 K_1^{-1} \mathcal{H}[\mu] & R_2 K_2^{-1} \mathcal{H}[\nu] \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = t_2 - t_1 \quad (83)$$

$$A \vec{x} = \vec{y}, \quad (84)$$

de este modo podemos obtener λ_1 y λ_2 mediante la solución de mínimos cuadrados descrita de la siguiente manera

$$\vec{x} = (A^T A)^{-1} A^T \vec{y}, \quad (85)$$

para así finalmente obtener el punto en el espacio mediante la siguiente ecuación

$$\mathbf{p} = \frac{1}{2}(t_1 + t_2 + \lambda_1 R_1 K_1^{-1} \mathcal{H}[\mu] + \lambda_2 R_2 K_2^{-1} \mathcal{H}[\nu]) \quad (86)$$

$$\mathbf{p} = \frac{1}{2}(t_1 + t_2 + \lambda_1 d_1 + \lambda_2 d_2). \quad (87)$$

3. Resultados

En el catálogo con realidad aumentada se incluyeron los siguientes productos:

- Caja de medicamento Pepto-Bismol
- Audífonos inalámbricos skullcandy sesh evo
- Calculadora FX-991EX Casio
- Candado de combinación SKG
- Raspberry pi 4

A continuación, la descripción de los resultados obtenidos se realizará mostrando únicamente la reconstrucción 3D de la calculadora casio FX-991EX.

3.1. Descripción del sistema

El sistema para la digitalización de objetos consta de un proyector Epson PowerLite W39 con una resolución de 1920x1080, una cámara Logitech C920 HD Pro Webcam de resolución 1920x1080, y un patrón de calibración color amarillo como se muestra en la Fig. 2.

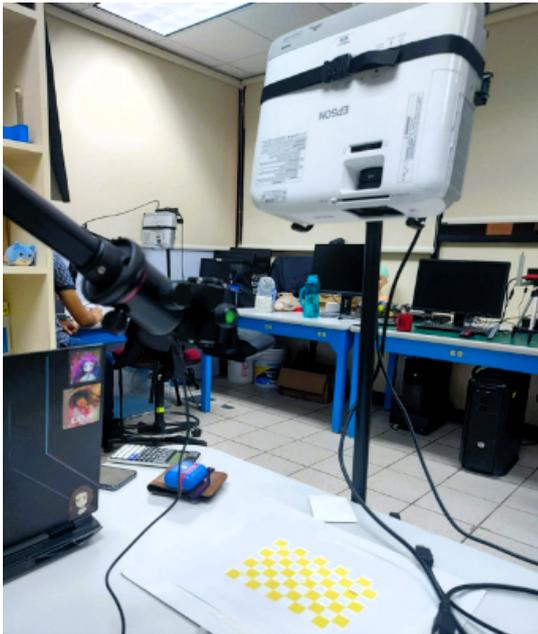


Figura 2: Sistema experimental de proyección de franjas para digitalización 3D.

3.2. Calibración

La calibración se realizó con ambos dispositivos al mismo tiempo, cámara y proyector. Se tomaron 5 imágenes con diferentes ángulos del patrón amarillo y del patrón cian proyectado encima y separando canales de colores podemos extraer ambos patrones de manera individual de una sola imagen.

Los parámetros intrínsecos resultantes son los siguientes:

$$K_c = \begin{bmatrix} 1,4419 & -0,0201 & 0,0358 \\ 0 & 1,4444 & 0,0999 \\ 0 & 0 & 1 \end{bmatrix}, \quad (88)$$

$$K_p = \begin{bmatrix} 3,4635 & 0,0091 & -0,1694 \\ 0 & 3,5083 & 0,3953 \\ 0 & 0 & 1 \end{bmatrix}, \quad (89)$$

donde K_c es la matriz de la cámara y K_p es la matriz del proyector.

De las 5 fotos tomadas para calibrar el sistema, se tomó la última para calcular los parámetros extrínsecos los cuales son los siguientes:

$$R_c = \begin{bmatrix} -0,9786 & -0,0704 & -0,1935 \\ -0,0166 & 0,9636 & -0,2670 \\ 0,2052 & -0,2581 & -0,9441 \end{bmatrix}, \quad (90)$$

$$R_p = \begin{bmatrix} 0,9599 & -0,0200 & -0,2797 \\ 0,0413 & -0,9764 & 0,2118 \\ -0,2774 & -0,2148 & -0,9364 \end{bmatrix}, \quad (91)$$

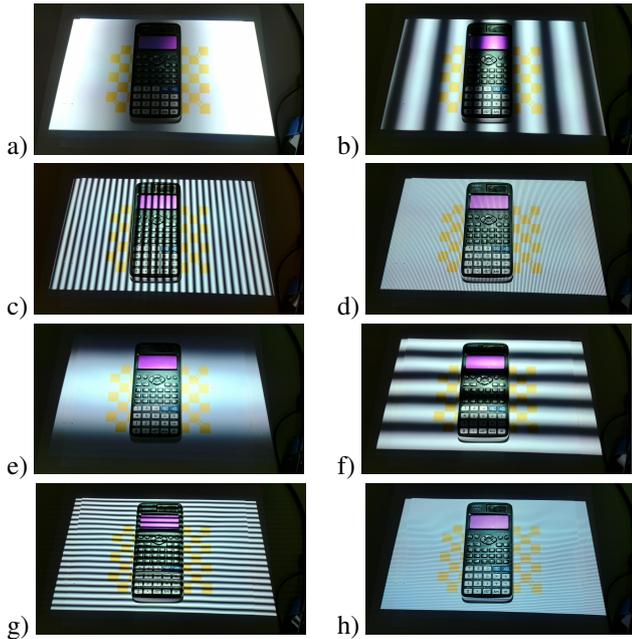
$$t_c = \begin{bmatrix} 23,3316 \\ 129,2237 \\ 266,6449 \end{bmatrix}, t_p = \begin{bmatrix} 113,3374 \\ -170,4995 \\ 490,8966 \end{bmatrix}, \quad (92)$$

donde R_c y t_c son la matriz de rotación y el vector de traslación, respectivamente, de la cámara, mientras que R_p y t_p son la matriz de rotación y el vector de traslación, respectivamente, del proyector.

Como nota para el lector todas las medidas de distancias se encuentran dadas en milímetros ya que así fue la manera en que se calibró el sistema, también puede ser el otras unidades pero se debe tener en cuenta al momento de medir el patrón de calibración.

3.3. Obtención de imágenes

En la generación de franjas para proyectar sobre el objeto se consideraron 16 corrimientos de fase con 4 tipos de rejillas, como se muestra en la Figura ??, dando un total de 64 imágenes, pero esto lo tenemos que hacer tanto para franjas horizontales como para franjas verticales.



Cuadro 1: Rejillas proyectadas con franjas verticales y horizontales con diferente frecuencia. Por cada rejilla, se realizaron 16 desplazamientos de fase.

3.4. Demodulación de fase

Posteriormente que ya hemos capturado las imágenes, obtenemos su fase desenvuelta, que en palabras sencillas, es cómo es la topología del objeto en la diapositiva que emite el proyector, como se muestra en la Figura 3.

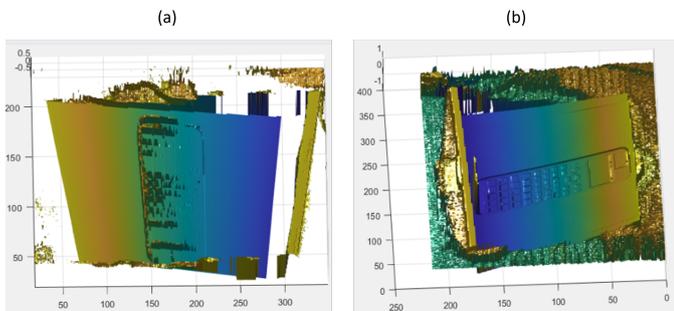


Figura 3: Resultado del proceso de demodulación de fase procesando (a) franjas horizontales, y (b) franjas verticales.

3.5. Nube de puntos

Después de obtener los mapas de fase del objeto, ya solo queda usar el método de triangulación entre dicha fase y las imágenes capturadas por la cámara, con este procedimiento obtenemos una nube de puntos que el proyector fue capaz de iluminar en el objeto, el resultado se muestra en la Fig. 4. Cabe mencionar que esta nube de puntos aún no es un objeto sólido listo para usar, pues solo son puntos y no tiene una superficie construida, además de que tiene mucho ruido que se debe eliminar.

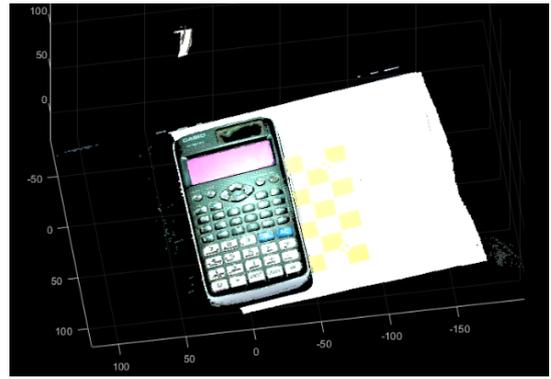


Figura 4: Nube de puntos del objeto obtenida al finalizar el proceso de digitalización.

3.6. Post procesamiento

Ya que hemos obtenido con éxito el objeto digitalizado con su nube de puntos, el siguiente paso es eliminar el ruido del modelo, que consiste en remover los puntos que no pertenecen al objeto. Para este proceso ocuparemos la herramienta “cloudcompare”, además, en esta misma aplicación generaremos la superficie del objeto (ya que hasta ahora es únicamente una nube de puntos) como en la figura 5(a) y 5(b).

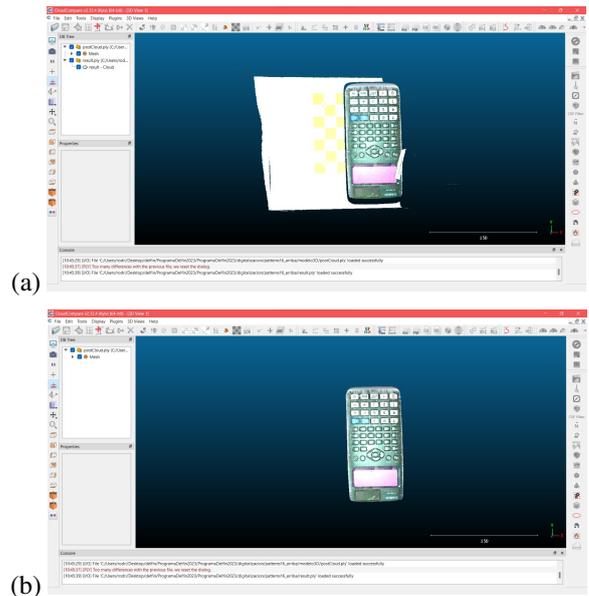


Figura 5: (a) Nube de puntos del objeto en cloudcompare para realizar la eliminación del ruido y la creación de la superficie. (b) Modelo 3D sin ruido y con una superficie.

Después de este procedimiento ya tenemos el modelo 3D del objeto con una superficie generada. Sin embargo, para exportar el objeto en la aplicación de Unity es necesario obtener su textura (para que el modelo cuente con color), esto se realizará en el programa MeshLab, con el que generaremos la textura del objeto. Además, tendremos que guardar el objeto en formato OBJ. El resultado de esto se muestra en la figura 6. Este procedimiento lo repetiremos dependiendo de las partes digitalizadas del objeto, en el caso de la calculadora serán dos veces (la par-

te frontal y la posterior), ya que tengamos todas las partes, las procederemos a unir manualmente en el programa de Blender, además de que reduciremos el número de polígonos con dicho programa, esto con la finalidad de que el modelo sea menos pesado para la aplicación móvil, como se puede apreciar en la figura 7(a).

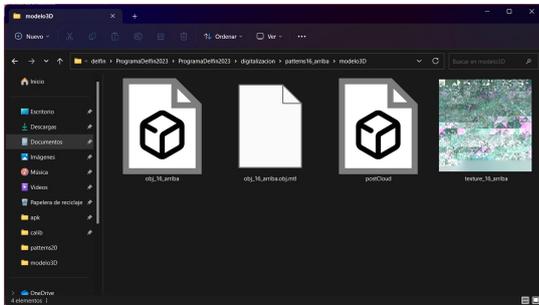
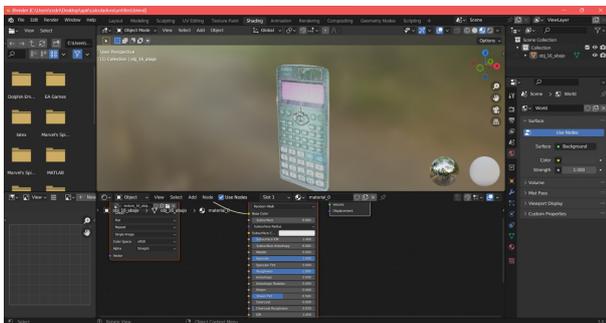
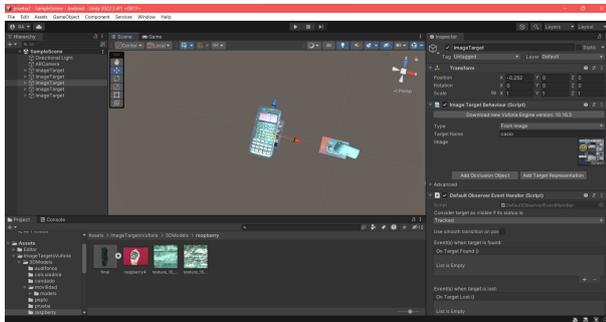


Figura 6: Modelo 3D del objeto en formato OBJ con sus texturas.



(a)

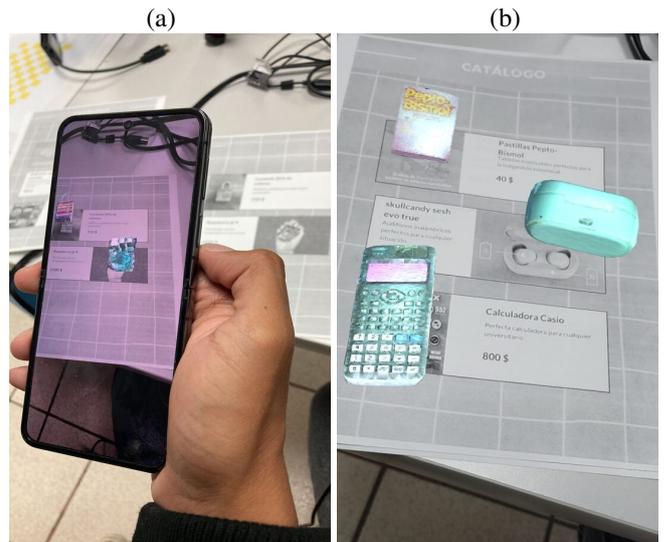


(b)

Figura 7: (a) Modelo 3D completo en el software Blender para unir las superficies parciales del objeto en una sola. (b) Objetos digitalizados en el programa de Unity para realizar la aplicación móvil con el uso de la librería Vuforia.

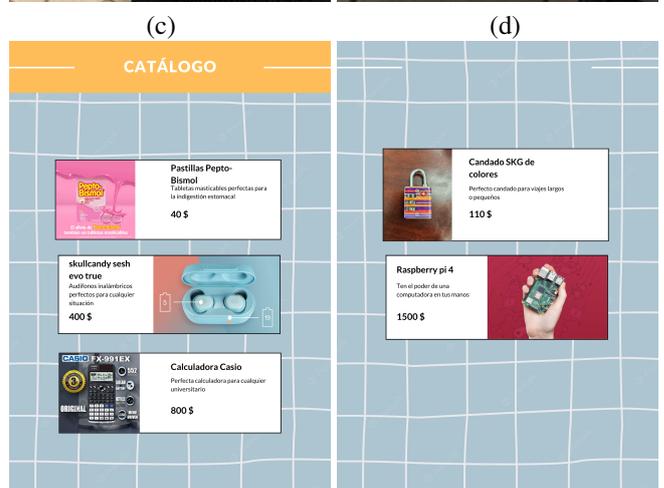
Después de unir las reconstrucciones parciales del objeto en una única reconstrucción de manera exitosa, exportaremos estos modelos a formato “FilmBox” (FBX) para usarlo en el programa de Unity y así generar la aplicación móvil de realidad aumentada como en la figura 7(b) con la ayuda de la librería Vuforia. Esta librería nos servirá para unir las imágenes objetivo con los modelos 3D obtenidos en la digitalización y diseñar el comportamiento de realidad aumentada que tendrá la aplicación.

Por último, se creó la aplicación móvil en el software Unity. En la Fig. 8 se puede apreciar el resultado final del proyecto empleando el catálogo empleado como demostración. La Fig. 9



(a)

(b)



(c)

(d)

Figura 8: Aplicación móvil en funcionamiento. (a) Resultado de la aplicación móvil donde la cámara del dispositivo captura el catálogo impreso y despliega en tiempo real el modelo 3D digital. (b) Captura de pantalla del dispositivo mientras captura dinámicamente los modelos 3D del catálogo impreso observado. (c) Catálogo usado para la aplicación móvil (a) Parte frontal del catálogo con las imágenes objetivo. (d) Parte posterior del catálogo con las imágenes objetivo.

muestra algunos de los objetos digitalizados empleados para construir el catálogo de muestra para ilustrar la operabilidad del proyecto desarrollado.

4. Conclusiones

En este proyecto se realizó con éxito la digitalización 3D de objetos mediante la proyección de franjas y se construyó una aplicación móvil orientada a la publicidad con realidad aumentada. Los modelos digitalizados se integraron en la aplicación para celular con android 8.0 o superior. En la aplicación se visualiza el modelo 3D a tamaño escala de los objetos digitalizados, esto se hace apuntando a imágenes específicas previamente cargadas.

Una de las ventajas del uso del sistema de proyección de franjas fue la reducción del tiempo de creación de los modelo

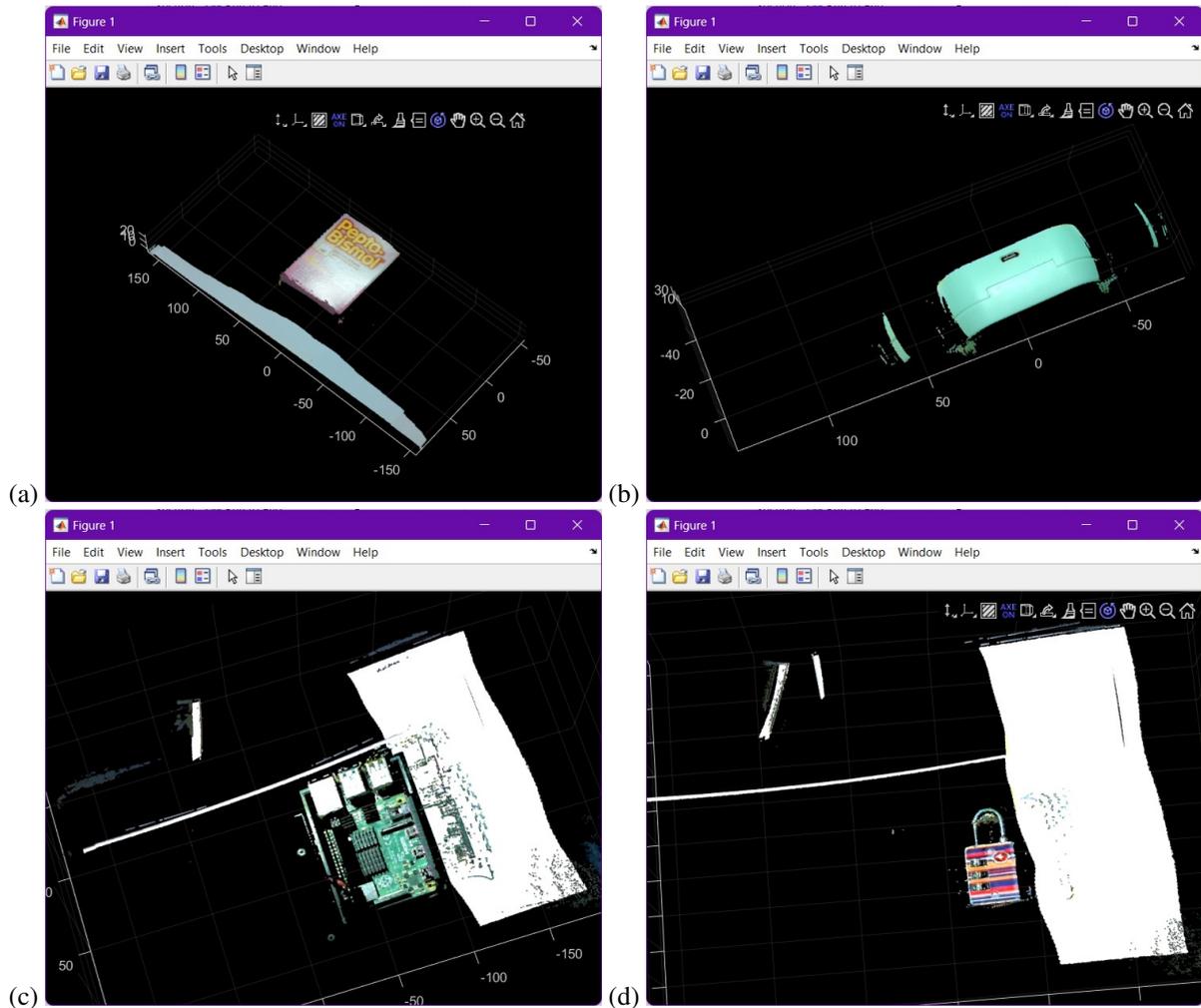


Figura 9: Nubes de puntos de los objetos digitalizados para demostración del proyecto. (a) Caja de medicina. (b) Audífonos inalámbricos. (c) Raspberry Pi 4. (d) Candado.

3D. Además, el sistema de proyección de franjas es sencillo de armar y económico, claro está que mientras mejor calidad tengan los dispositivos empleados mejor será la digitalización.

Las complicaciones de emplear esta técnica son que no existe un procedimiento eficiente para generar los modelos 3D, ya que fue necesario mucho trabajo de post procesamiento y perjudica la calidad de los objetos digitalizados. Además, debido al gran número de características que se obtienen del objeto, los modelos 3D requieren mucho espacio de almacenamiento.

Como mejoras en trabajo a futuro se propone la creación de un método que genere los modelos 3D de forma automática, así como una optimización en los mismos para reducir el espacio que ocupan y generar formas de interacción en la aplicación de realidad aumentada.

Referencias

- [1] R. Juárez-Salazar and V. Díaz-Ramírez, "Operator-based homogeneous coordinates: Application in camera document scanning," *Optical Engineering*, vol. 56, July 2017. Publisher Copyright: © 2017 Society of Photo-Optical Instrumentation Engineers (SPIE).
- [2] R. Juárez-Salazar, C. Mendoza-Rodríguez, J. E. Hernández-Beltrán, and C. Robledo-Sánchez, "How do phase-shifting algorithms work?," *European Journal of Physics*, vol. 39, p. 065302, oct 2018.
- [3] R. Juárez-Salazar, F. Guerrero-Sánchez, and C. Robledo-Sánchez, "Theory and algorithms of an efficient fringe analysis technology for automatic measurement applications," *Appl. Opt.*, vol. 54, pp. 5364–5374, Jun 2015.
- [4] R. Juárez-Salazar, A. Giron, J. Zheng, and V. H. Díaz-Ramírez, "Key concepts for phase-to-coordinate conversion in fringe projection systems," *Applied Optics*, vol. 58, pp. 4828–4834, jun 2019.
- [5] R. Juárez-Salazar, S. Esquivel-Hernández, and V. H. Díaz-Ramírez, "Are camera, projector, and camera-projector calibrations different?," *Appl. Opt.*, vol. 62, pp. 5999–6006, Aug 2023.
- [6] R. Juárez-Salazar and V. H. Díaz-Ramírez, "Flexible camera-projector calibration using superposed color checkerboards," *Optics and Lasers in Engineering*, vol. 120, pp. 59–65, 2019.
- [7] R. Juárez-Salazar, G. A. Rodríguez-Reveles, S. Esquivel-Hernández, and V. H. Díaz-Ramírez, "Three-dimensional spatial point computation in fringe projection profilometry," *Optics and Lasers in Engineering*, vol. 164, p. 107482, May 2023.