




# MODELO DE FORMACIÓN DE LA IMAGEN

PEDRO JOSÉ DÍAZ GARCÍA Y RAÚL OLMO RUIZ

SISTEMAS DE PERCEPCIÓN, 4º GIERM  
Ejercicio Práctico 1



El objetivo de este ejercicio es la realización de un programa en MATLAB capaz de representar un objeto y la imagen que se obtendría del mismo en función de la posición, orientación y parámetros de una cámara. Esta memoria está estructurada de la siguiente manera: en primer lugar, se explica las distintas partes que conforman el código del programa, y posteriormente se lleva a cabo un análisis de los resultados obtenidos al realizar posibles variaciones en los parámetros variables.

# 1. Exposición del programa

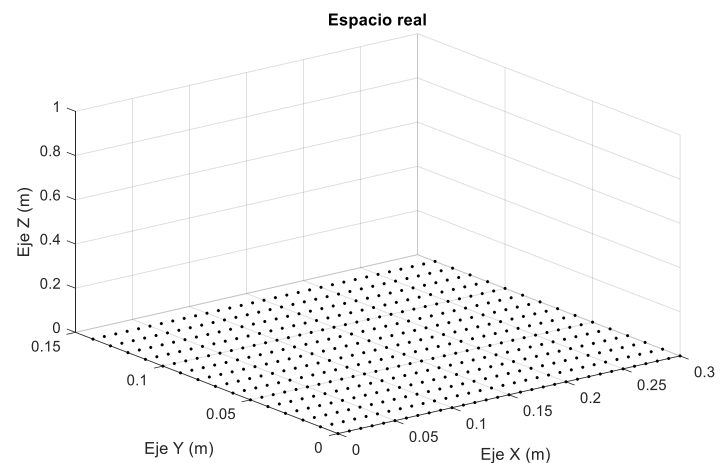
El programa está estructurado en tres bloques: el primero lleva a cabo la definición del objeto a representar y de las condiciones en las que se realiza la adquisición de la imagen (parámetros de la cámara, posición y orientación relativas al objeto); el segundo bloque se encarga de el cálculo necesario para obtener las coordenadas de los puntos representados en el plano de la imagen; y el tercer y último bloque se encarga de hacer la representación gráfica de lo obtenido en los bloques anteriores, esto es: objeto, sistema de referencia del objeto, sistema de referencia de la cámara y plano de la imagen. Cabe destacar que el segundo y tercer bloque se encuentran dentro de un bucle, cuya única finalidad es la representación continuada de los resultados obtenidos al variar la posición y orientación de la cámara. A continuación, se indagará más en cada uno de estos bloques.

## 1.1. Bloque 1: Definición del objeto, parámetros de cámara y posición y orientación.

El objeto definido se trata de un conjunto rectangular de puntos, todos ellos distribuidos en el plano x-y. Para ello, se da la posibilidad de elegir el número de puntos que se quiere en cada lado del rectángulo, la distancia que debe existir entre una pareja de puntos y la posición del origen del sistema de referencia del objeto. En las siguientes líneas de código, tras la definición de los datos anteriores, se realiza una ampliación sucesiva de una matriz para obtener finalmente una matriz en la que cada columna se corresponde con las coordenadas de uno de los puntos. Como puede comprobarse, el origen del sistema de referencia siempre estará situado en la esquina inferior izquierda del rectángulo.

```
P=[];  
for i=1:num_px  
    for j=1:num_py  
        P=[P [wx+(i-1)*dx;wy+(j-1)*dy;wz]];  
    end  
end
```

Seguidamente se definen todos los parámetros relativos a la cámara. Únicamente cabe recalcar la definición de la denominada matriz de parámetros intrínsecos, que se mantendrá constante independientemente de la posición y orientación posteriores que se le de a la cámara.



**Figura 1. Objeto: matriz de puntos.**

Por último, para el caso de la definición de la posición y orientación relativas de la cámara respecto al objeto, se ha optado por predeterminar nueve de estas configuraciones, de manera que en el bucle posterior se use una de ellas en cada iteración. A continuación, a modo de ejemplo, se muestra el código correspondiente a una de las configuraciones, en el que básicamente se lleva a cabo la composición de la matriz de transformación homogénea “cámara-a-mundo” a partir del vector traslación y la matriz de rotación que definen la posición de los ejes de la cámara respecto del sistema de referencia del objeto; y se usan dos matices tridimensionales para almacenar cada una de estas matrices de transformación y sus inversas. Esto es así por que las primeras serán útiles para la representación gráfica de los ejes de la cámara, y las segundas son necesarias para el cálculo de proyección de los puntos.

```
%1_Distancia media y enfoque recto hacia el objeto (inicial).
wTc=[round((num_px-1)/2)*dx,round((num_py-1)/2)*dy,0.4]'; %Vector traslación de
{C} respecto {W}
Rx=[1 0 0;0 cos(pi) -sin(pi);0 sin(pi) cos(pi)];
wRc=Rx; %Matriz de rotación de {C} respecto {W}
wTc=[wRc wTc;0 0 0 1]; %Matriz de parámetros extrínsecos (matriz de
transformación homogénea)
WTC(:, :, 1)=wTc; %Almacenamiento de las transformaciones homogéneas {W}-{C}
cTw=inv(wTc);
CTW(:, :, 1)=cTw; %Almacenamiento de las transformaciones homogéneas {C}-{W}
```

## 1.2. Bloque 2: Cálculo de la proyección de los puntos.

En el segundo bloque, se realizan los cálculos necesarios para conocer las coordenadas, expresadas en píxeles, de cada punto del objeto en el plano de la imagen, de manera que pueda representarse el conjunto completo de puntos captados por la cámara. En este caso, se dan dos posibilidades, en función de si se quiere tener en cuenta el efecto de distorsión de la lente o no; esto quedará determinado al comienzo de la ejecución del programa.

En el caso más básico, es decir, sin distorsión, teniendo ya definida la matriz de parámetros intrínsecos, la matriz de parámetros extrínsecos y todas las coordenadas de los puntos del objeto agrupadas en una misma matriz, el cálculo necesario se reduce a tres pasos (que puede verse en las siguientes líneas de código): paso de las coordenadas reales del objeto a coordenadas homogéneas, transformación de dichas coordenadas al plano de la imagen (ya expresadas en píxeles), y devolver estas coordenadas reales.

```
%Se hacen homogéneas las coordenadas de los puntos del objeto
P_=[P;ones(1,size(P,2))];
%Cálculo sin tener en cuenta la distorsión
p_ = A*CTW(1:3, :, j)*P_;
for i=1:size(p_,2)
    p(:,i)=p_(1:2,i)/p_(3,i);
end
```

En el caso de que decida tener en cuenta la distorsión de lente, el proceso de cálculo se amplía. El proceso genérico para este cálculo pasa por realizar el mismo procedimiento de tres pasos descrito en el párrafo anterior, pero incluyendo una etapa intermedia en la que se incluya el efecto de la distorsión. Sin embargo, con el fin de aprovechar el cálculo de los puntos proyectados ya realizado, en este código se ha optado por volver a expresar dichas coordenadas en m, aplicarles las correspondientes componentes de distorsión y volver a expresarlas en píxeles. Esto puede verse en las siguientes líneas de código.

```
%Paso de las coordenadas calculadas anteriormente (en píxeles)
%a coordenadas en m, y normalizar.
pn=[1/f*rox 0;0 1/f*roy]*(p-[u0;v0]);

%Cálculo de posiciones distorsionadas
for i=1:size(pn,2)
    pd(:,i)=pn(:,i)*(1+kr1*norm(pn(:,i))^2+kr2*norm(pn(:,i))^4)+...
        +[2*kt1*pn(1,i)*pn(2,i)+kt2*(norm(pn(:,i))^2+2*pn(1,i)^2);
        2*kt2*pn(1,i)*pn(2,i)+kt1*(norm(pn(:,i))^2+2*pn(2,i)^2)];
end

%Se deshace el paso realizado previamente a la aplicación de la
%distorsión
p=[f/rox 0;0 f/roy]*pd+[u0;v0];
```

### 1.3. Bloque 3: Representación gráfica de los resultados.

Respecto al último de los bloques, no hay mucho que destacar, ya que se basa en su mayor parte en el uso de comandos para la representación de los cálculos realizados en los bloques anteriores. Para cada una de las configuraciones espaciales definidas, se verá una imagen con dos gráficas: a la izquierda estará representada la matriz de puntos del objeto y su sistema de referencia, así como los ejes de la cámara en su correcta configuración; a la derecha, aparecerá el plano de la imagen con la proyección de los puntos y un marco que representa los límites del sensor.

Como puede verse en la parte de código correspondiente, la única operación que se sale de la mera representación se encuentra en la representación de los ejes de la cámara, para la que se ha utilizado las matrices WTC, que se almacenaron durante la definición de las configuraciones, para obtener la disposición de dichos ejes con respecto a los del objeto.

Como última mención, la clave de colores que se ha usado para la representación de ejes es: rojo para el eje X, verde para el eje Y, azul para el eje Z.

## 2. Análisis de los resultados

En este apartado se quiere analizar los resultados que se obtienen empleando el programa que se ha descrito. Más concretamente, interesa ver cómo varía la proyección final de los puntos del objeto a medida que se modifican diversas variables, como son la posición y orientación de la cámara o los parámetros de la misma. Con el fin de facilitar la apreciación de estas variaciones en los resultados, se proponen diferentes experimentos en los que se modifiquen diferentes aspectos.

### 2.1. Modificación de la posición y de la orientación de la cámara.

Haciendo uso de todas las configuraciones espaciales que se han predefinido para la cámara, se obtiene una secuencia en la que poder apreciar cómo varía la proyección del objeto en el sensor en función de la posición y la orientación de la cámara respecto a él. A continuación, se exponen las diferentes representaciones obtenidas, para las que no se han modificado los parámetros del sensor ni se ha tenido en cuenta la distorsión de la lente.

En primer lugar, conviene atender a la siguiente imagen, ya que corresponde a la primera de las configuraciones y servirá de referencia para apreciar la variación en las posteriores configuraciones. En esta, la cámara se encuentra situada justo encima del centro del objeto, orientada de tal manera que el eje de la cámara y el plano del objeto resulten perpendiculares.

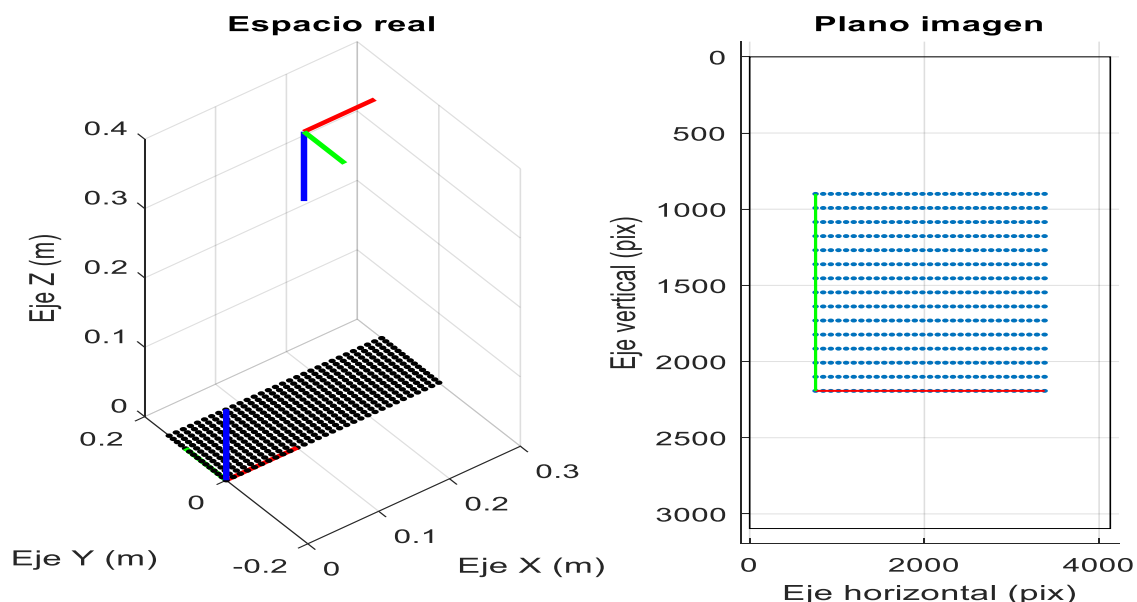


Figura 2. Config. 1: referencia inicial.

La primera modificación que se ha llevado a cabo es realizar un acercar y alejar la cámara del objeto manteniendo la orientación de la posición anterior. Como puede apreciarse, en el caso de acercar la cámara, el sensor no alcanza a captar todos los puntos del objeto, excediendo algunos los límites de este. De forma análoga, si se aleja la cámara, aparece representado el objeto completo pero ocupando una porción menor del sensor, ya que ahora puede captar un espacio mucho más amplio.

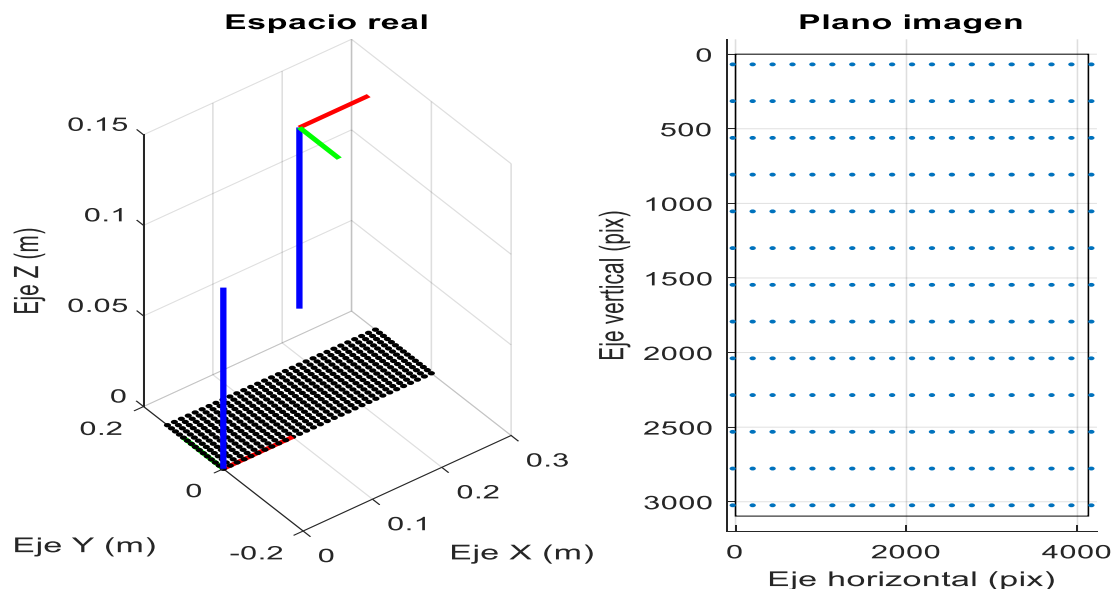


Figura 3. Config. 2: acercamiento.

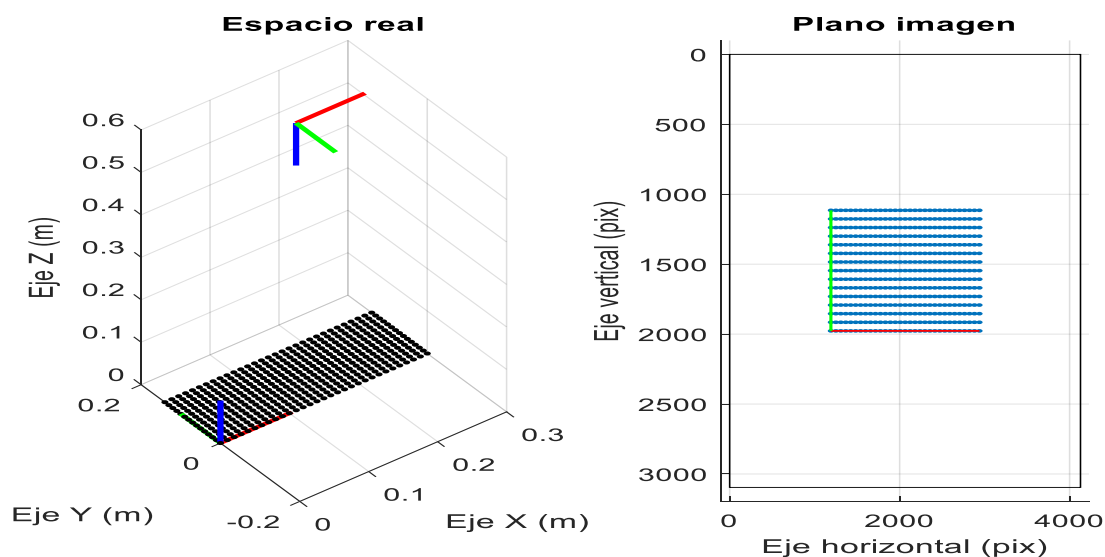


Figura 4. Config. 3: alejamiento.

El siguiente paso ha sido cambiar la posición de la cámara de manera que ya no estuviese sobre el centro del objeto, aunque aún se mantiene la misma orientación. El objetivo es ver que solo aparece representada la parte del objeto que capte el sensor y que esta se corresponde con la zona del objeto en la que la cámara se encuentre.

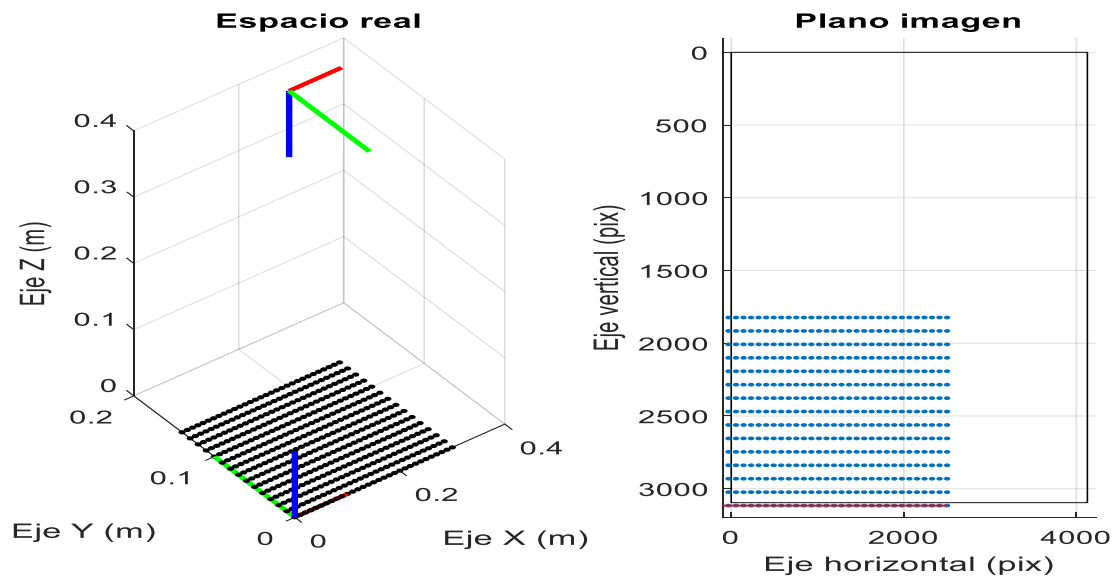


Figura 5. Config. 4: desplazamiento.

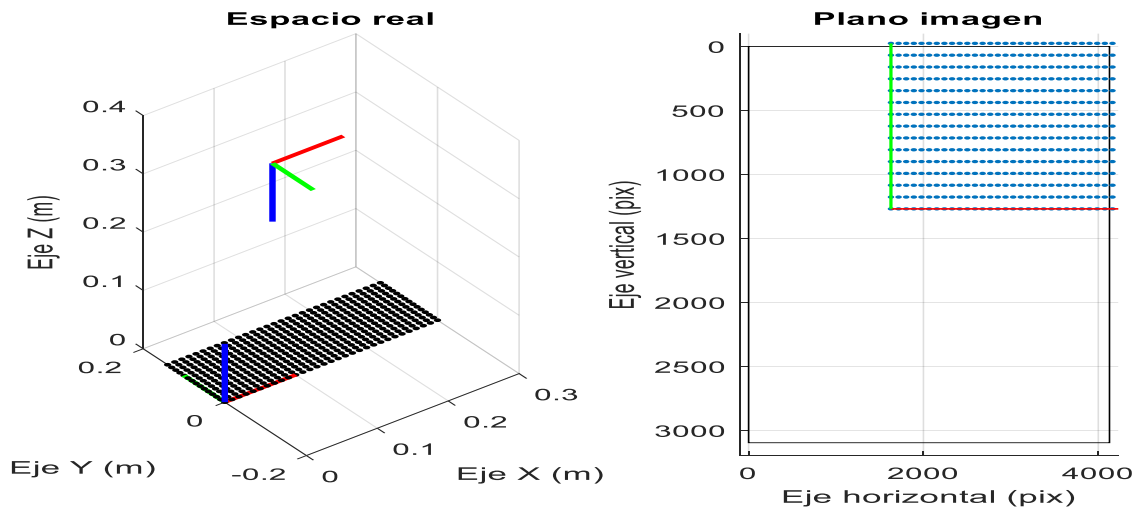


Figura 6. Config. 5: desplazamiento.

Como caso extremo de esto último, se ha desplazado la cámara tanto que le es imposible captar el objeto (al menos con la misma orientación). Este resultado puede apreciarse en la figura siguiente, donde obviamente no aparece nada representado en el plano de la imagen.

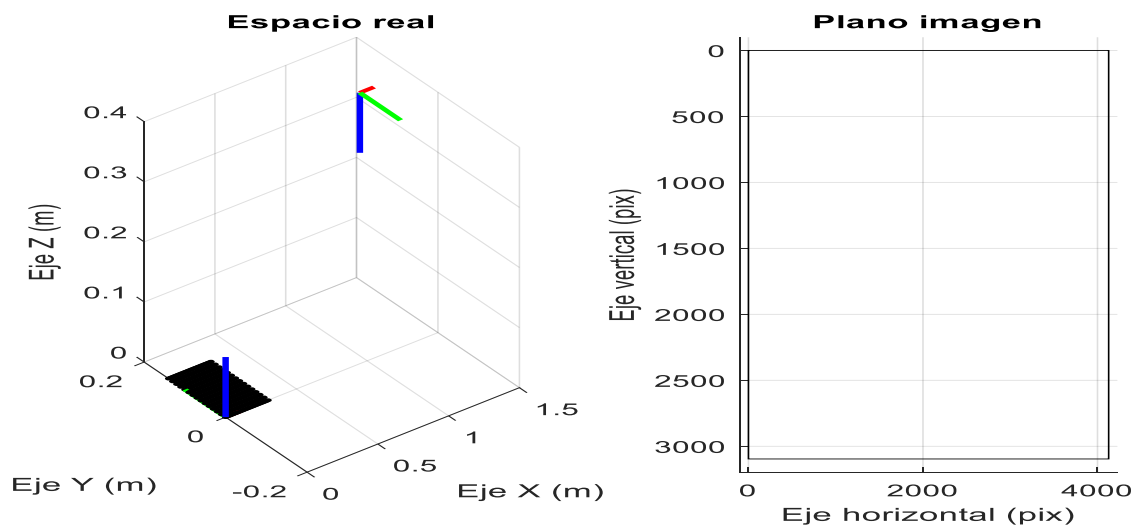


Figura 7. Config. 6: desplazamiento lejano.

Volviendo a la posición inicial (configuración 1), se va a obtener la proyección al variar la orientación de la cámara pero manteniéndola situada sobre el centro del objeto. En la figura 8 puede verse el resultado de rotar el sensor un ángulo recto manteniendo el enfoque perpendicular hacia el objeto; para obtener la figura 9, se ha optado por variar la inclinación, de forma que el enfoque ya no resulta perpendicular.

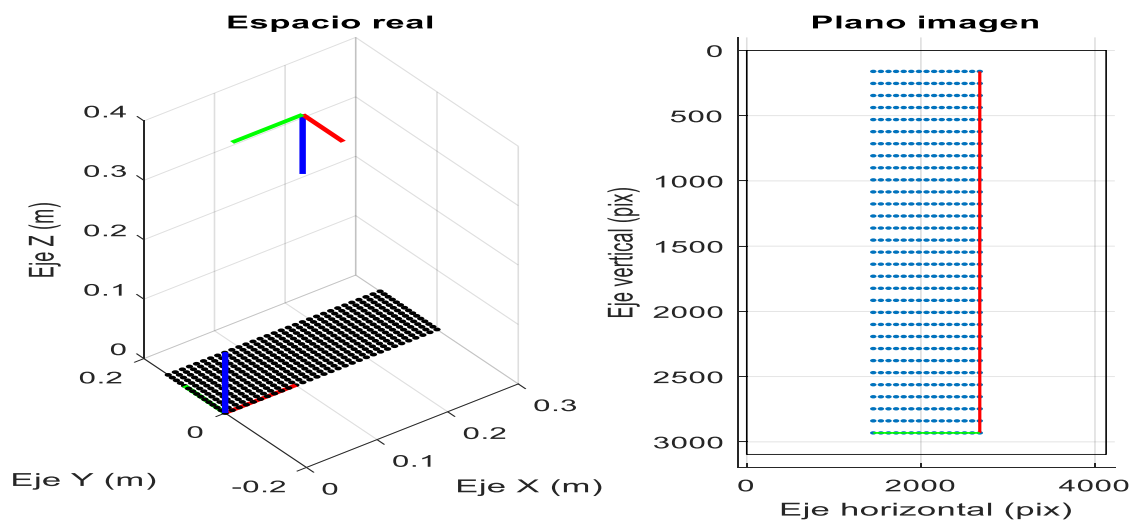


Figura 8. Config. 7: rotación respecto al eje de la cámara.

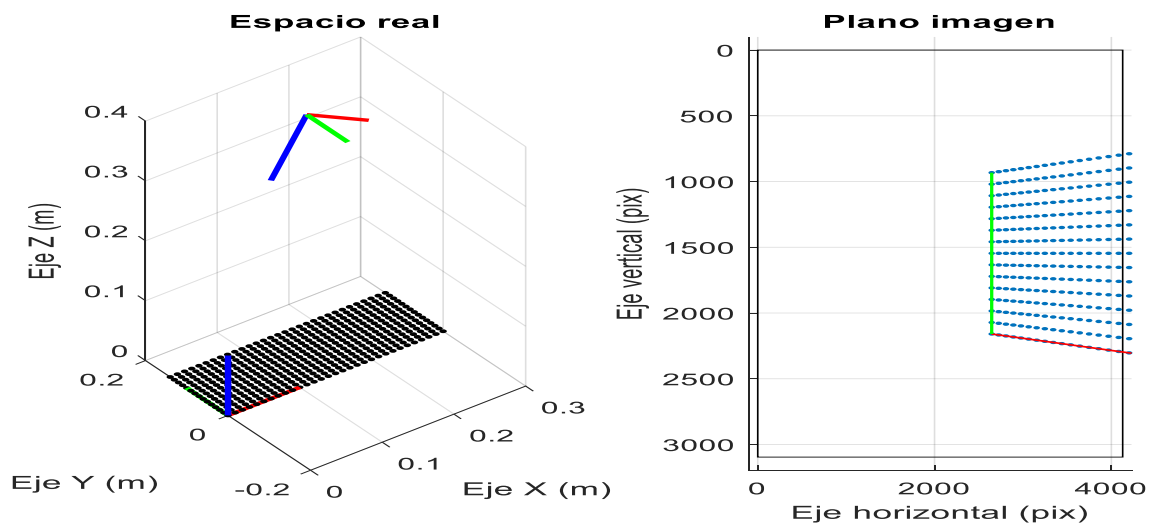


Figura 9. Config. 8: inclinación.

Por último, se ha querido representar una configuración que haya sufrido tanto desplazamiento como cambio de orientación con respecto a la configuración inicial, de manera que se aprecie el efecto combinado sobre el plano de la imagen.

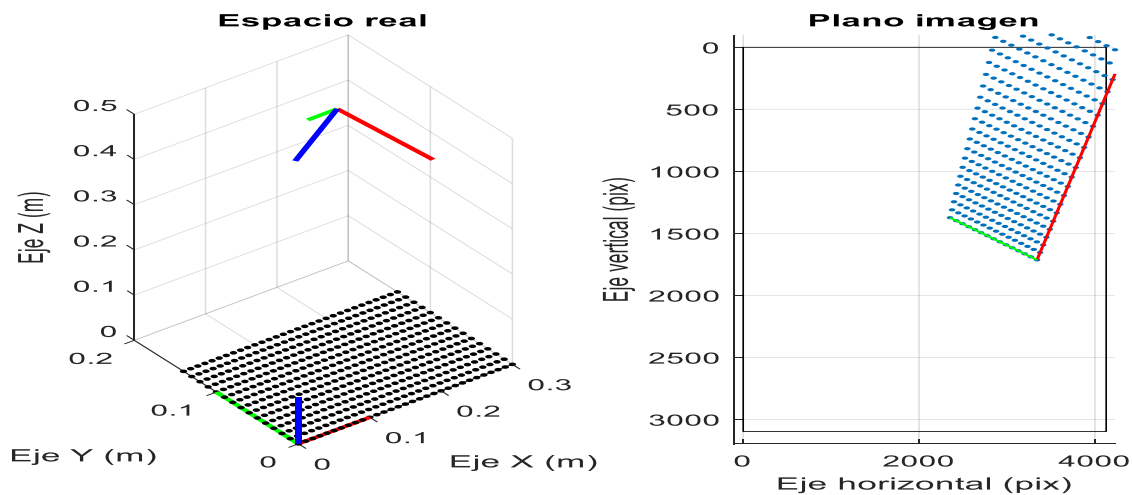


Figura 10. Config. 9: desplazamiento y cambio de orientación combinados.

## 2.2. Efecto de la distorsión de la lente.

Para apreciar el efecto de distorsión, se utilizará la primera configuración, ya que se considera que es la mejor para comprobar cómo afecta la variación de los parámetros de distorsión a la proyección de la imagen. La proyección resultante tomando los parámetros del sensor dados está representada en la figura 11, aunque el efecto final no sea muy apreciable. Por ello, se va a variar el valor de los parámetros de distorsión, para así poder apreciar el efecto que tienen sobre la imagen.

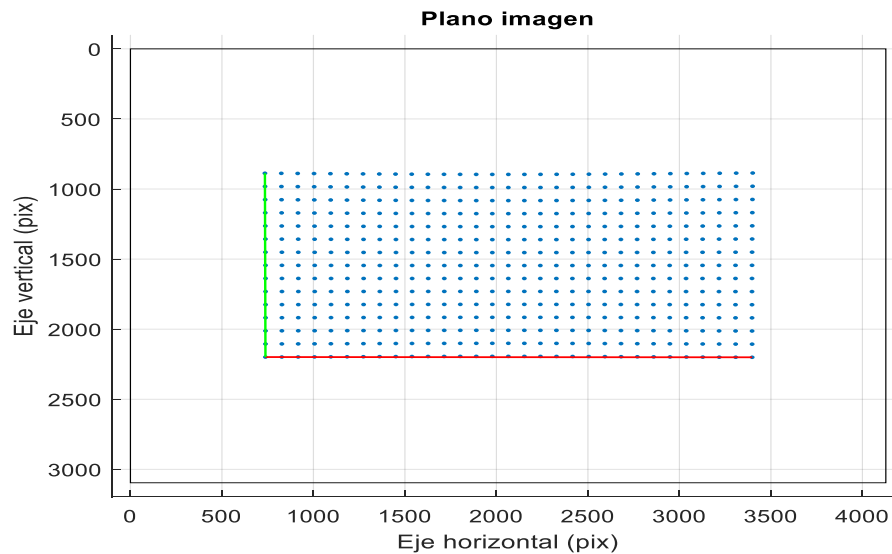


Figura 11. Efecto de distorsión sobre la proyección ( $kr_1=0.144$ ,  $kr_2=-0.307$ ,  $kt_1=-0.0032$ ,  $kt_2=0.0017$ ).

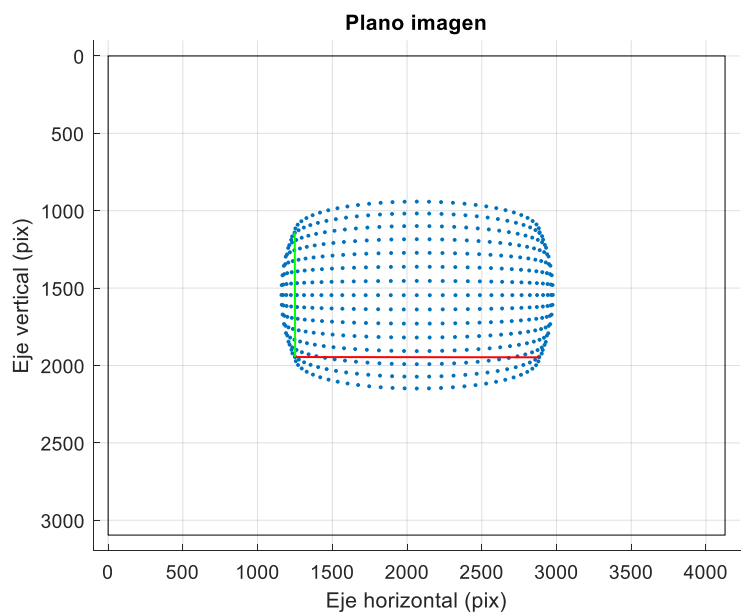
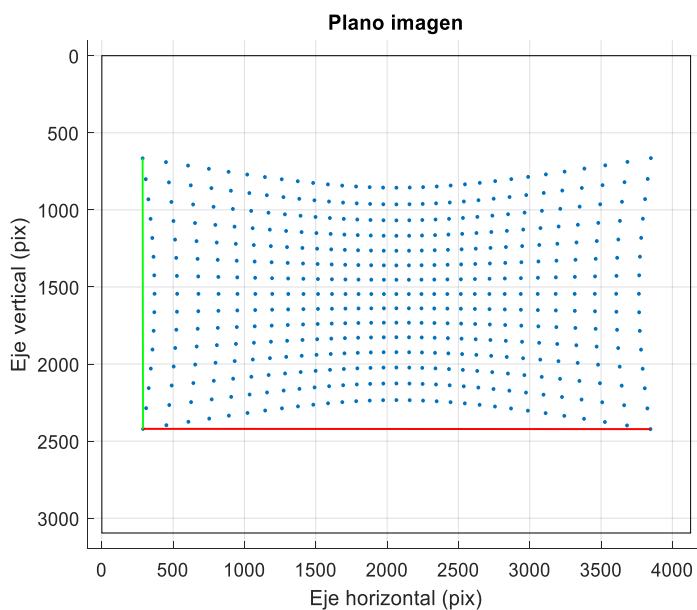
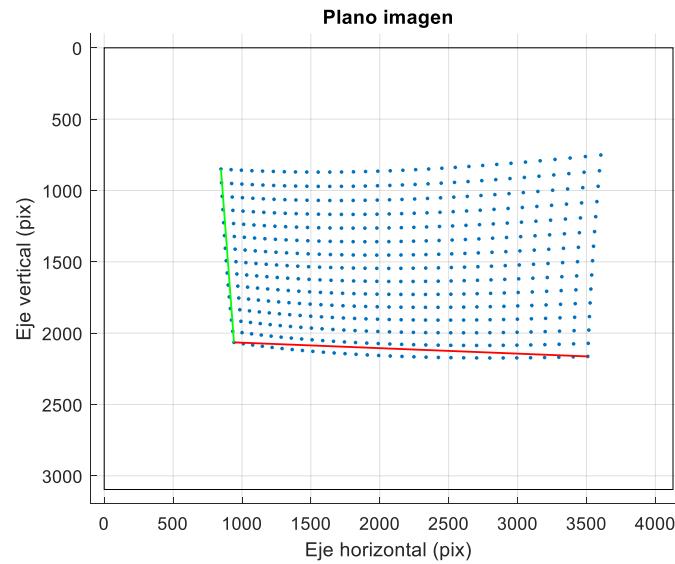


Figura 12. Izquierda ( $kr_1=2.144$ ,  $kr_2=-0.307$ ), derecha ( $kr_1=-2.144$ ,  $kr_2=-0.307$ ).

Como puede comprobarse, un aumento de la magnitud del parámetro de distorsión hace esta más evidente. Además, en función del signo del parámetro, la distorsión se produce de forma que los puntos se alejan unos de otros conforme más alejados se encuentran del centro de la imagen (valor positivo) o, por el contrario, se aproximan entre ellos (valor negativo).

Lo comentado anteriormente se corresponde con la distorsión radial. Por su parte, la distorsión tangencial suele despreciarse frente a la radial, ya que su efecto es notablemente menor. Sin embargo, se ha querido hacer notar el efecto de esta, por lo que también se han incrementado un poco los parámetros correspondientes.

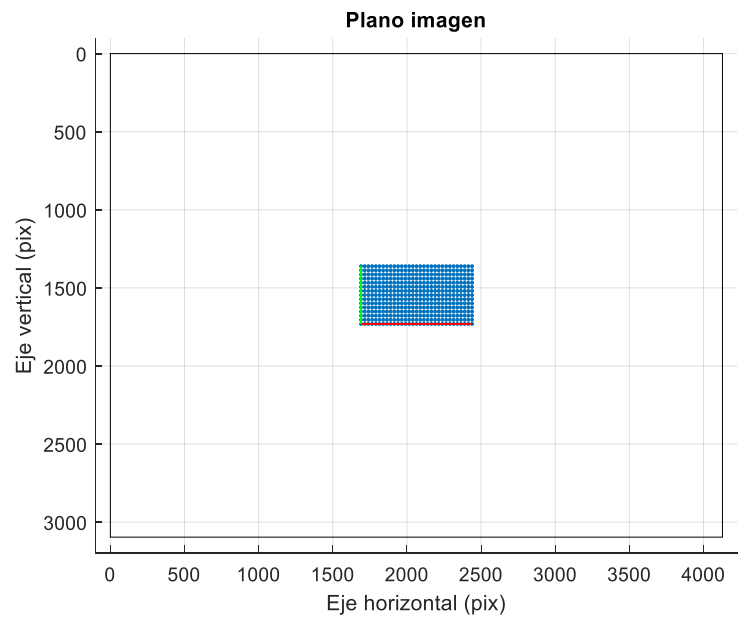
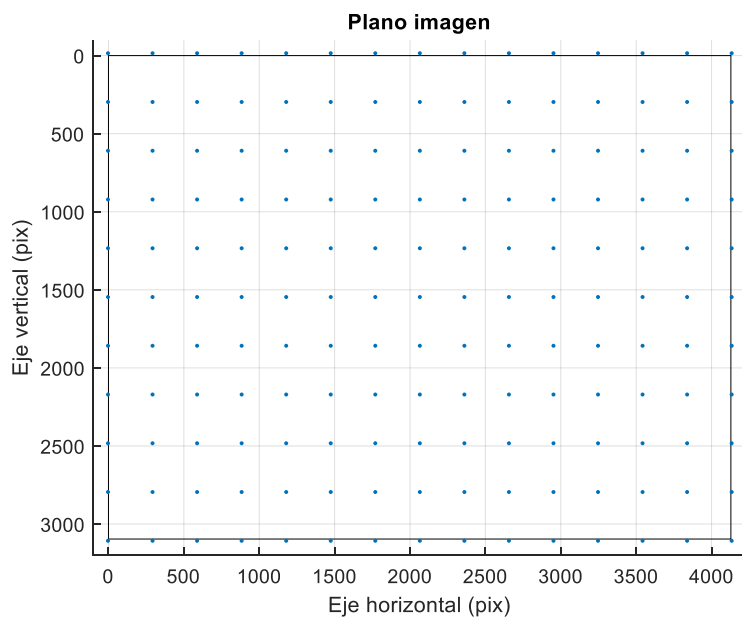




**Figura 13. Efecto de distorsión sobre la proyección ( $kr1=0.144$ ,  $kr2=-0.307$ ,  $kt1=-0.1032$ ,  $kt2=0.1017$ ).**

### 2.3. Variación de la distancia focal.

Por último, se ha querido comprobar la diferencia de resultados en función del valor de la distancia focal de la lente. Nuevamente, se utilizará como referencia los resultados obtenidos con la configuración 1. Como puede comprobarse, un valor mayor de distancia focal hace que la proyección de los puntos se disperse más, por lo que el plano de la imagen no alcanza a captar todos ellos. De igual manera, si se disminuye la distancia focal, los puntos aparecerán más concentrados.



**Figura 14. Variación de la distancia focal. Izquierda ( $f=0.0142$ ), derecha ( $f=0.0012$ ).**

# Anexo: código completo.

%BLOQUE 1: DEFINICIÓN DEL OBJETO, DE LOS PARÁMETROS DE LA CÁMARA Y DE LA CONFIGURACIÓN ESPACIAL DE LA CÁMARA RESPECTO DEL OBJETO.

%Al ejecutar el programa, se le pide al usuario que decida si quiere tener en cuenta el efecto de la distorsión o no.

```
distor=input('¿Tener en cuenta la distorsión? Yes:1 No:0\n');
```

%DEFINICIÓN DEL OBJETO

%Datos relativos al objeto.

num\_px=31; %Número de puntos en el eje horizontal

num\_py=15; %Número de puntos en el eje vertical

dx=0.01; %Separación(en m) entre puntos en el eje horizontal

dy=0.01; %Separación(en m) entre puntos en el eje vertical

wx=0;wy=0;wz=0; %Coordenadas del origen del sistema de referencia del objeto {W}

%Matriz que contiene las coordenadas de todos los puntos del objeto

% [ x1 x2 x3 . . . . xn]

% [ y1 y2 y3 . . . . yn]

% [ z1 z2 z3 . . . . zn]

P=[];

for i=1:num\_px

for j=1:num\_py

P=[P [wx+(i-1)\*dx;wy+(j-1)\*dy;wz]];

end

end

%PARÁMETROS DE LA CÁMARA

%Datos relativos a la cámara

f=0.0042; %Distancia focal (m)

N=4128; M=3096; %Resolución de la imagen (N:ancho, M:alto) (pix)

w=0.00496; h=0.00352; %Tamaño del sensor (w:ancho, h:alto) (m)

u0=round(N/2)+1; v0=round(M/2)-2; %Punto principal de la imagen (pix)

s=0; %skew

kr1=0.144; kr2=-0.307; %Coeficientes de distorsión radial

kt1=-0.0032; kt2=0.0017; %Coeficientes de distorsión tangencial

%Cálculo de matriz de parámetros intrínsecos

rox=w/N; %Dimensión efectiva del píxel en el eje horizontal

roy=h/M; %Dimensión efectiva del píxel en el eje vertical

A=[f/rox s\*f/rox u0;0 f/roy v0;0 0 1]; %Matriz de parámetros intrínsecos

%CONFIGURACIÓN ESPACIAL DE LA CÁMARA

%Declaración de una serie de configuraciones predeterminadas, pensadas

%para el posterior análisis de resultados.

%1\_Distancia media y enfoque recto hacia el objeto (inicial).

wtc=[round((num\_px-1)/2)\*dx,round((num\_py-1)/2)\*dy,0.4]'; %Vector traslación de

{C} respecto {W}

Rx=[1 0 0;0 cos(pi) -sin(pi);0 sin(pi) cos(pi)];

wRc=Rx; %Matriz de rotación de {C} respecto {W}

wTc=[wRc wtc;0 0 0 1]; %Matriz de parámetros extrínsecos (matriz de

transformación homogénea)

WTC(:, :, 1)=wTc; %Almacenamiento de las transformaciones homogéneas {W}-{C}

cTw=inv(wTc);

CTW(:, :, 1)=cTw; %Almacenamiento de las transformaciones homogéneas {C}-{W}

%2\_Acercamiento, enfoque recto.

wtc=[round((num\_px-1)/2)\*dx,round((num\_py-1)/2)\*dy,0.15]';

Rx=[1 0 0;0 cos(pi) -sin(pi);0 sin(pi) cos(pi)];

wRc=Rx;

wTc=[wRc wtc;0 0 0 1];

```
WTC(:, :, 2) = wTc;
cTw = inv(wTc);
CTW(:, :, 2) = cTw;
```

### %3\_Alejamiento, enfoque recto.

```
wtc = [round((num_px-1)/2)*dx, round((num_py-1)/2)*dy, 0.6]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
wRc = Rx;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 3) = wTc;
cTw = inv(wTc);
CTW(:, :, 3) = cTw;
```

### %4\_Desplazamiento lateral, enfoque recto.

```
wtc = [round((num_px-1)/2)*dx+0.1, round((num_py-1)/2)*dy+0.1, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
wRc = Rx;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 4) = wTc;
cTw = inv(wTc);
CTW(:, :, 4) = cTw;
```

### %5\_Desplazamiento lateral hacia el otro lado, enfoque recto.

```
wtc = [round((num_px-1)/2)*dx-0.1, round((num_py-1)/2)*dy-0.1, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
wRc = Rx;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 5) = wTc;
cTw = inv(wTc);
CTW(:, :, 5) = cTw;
```

### %6\_Desplazamiento lejano.

```
wtc = [round((num_px-1)/2)*dx+1, round((num_py-1)/2)*dy, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
wRc = Rx;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 6) = wTc;
cTw = inv(wTc);
CTW(:, :, 6) = cTw;
```

### %7\_Posición inicial con giro alrededor del eje de la cámara.

```
wtc = [round((num_px-1)/2)*dx, round((num_py-1)/2)*dy, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
Rz = [cos(pi/2) -sin(pi/2) 0; sin(pi/2) cos(pi/2) 0; 0 0 1];
wRc = Rx*Rz;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 7) = wTc;
cTw = inv(wTc);
CTW(:, :, 7) = cTw;
```

### %8\_Posición inicial con inclinación.

```
wtc = [round((num_px-1)/2)*dx, round((num_py-1)/2)*dy, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
Ry = [cos(-pi/6) 0 sin(-pi/6); 0 1 0; -sin(-pi/6) 0 cos(-pi/6)];
wRc = Rx*Ry;
wTc = [wRc wtc; 0 0 0 1];
WTC(:, :, 8) = wTc;
cTw = inv(wTc);
CTW(:, :, 8) = cTw;
```

### %9\_Desplazamiento y cambio arbitrario de orientación combinados.

```
wtc = [round((num_px-1)/2)*dx+0.1, round((num_py-1)/2)*dy+0.1, 0.4]';
Rx = [1 0 0; 0 cos(pi) -sin(pi); 0 sin(pi) cos(pi)];
Ry = [cos(-pi/5) 0 sin(-pi/5); 0 1 0; -sin(-pi/5) 0 cos(-pi/5)];
Rz = [cos(2*pi/5) -sin(2*pi/5) 0; sin(2*pi/5) cos(2*pi/5) 0; 0 0 1];
wRc = Rx*Ry*Rz;
```

```

wTc=[wRc wtc;0 0 0 1];
WTC(:, :, 9)=wTc;
cTw=inv(wTc);
CTW(:, :, 9)=cTw;

```

## %BLOQUE 2: CÁLCULO DE LA PROYECCIÓN DE LOS PUNTOS EN EL PLANO DE LA IMAGEN

```

%Se hacen homogéneas las coordenadas de los puntos del objeto
P_=[P;ones(1,size(P,2))];

%Bucle que recorre las configuraciones definidas anteriormente; incluye
%cálculo de la proyección correspondiente y su representación.
for j=1:9
    %Se espera a que el usuario pida ver la siguiente posición
    fprintf('Pulsar intro para ver el paso %d \n',j)
    pause;

    %Cálculo sin tener en cuenta la distorsión
    p_=A*CTW(1:3,:,j)*P_;
    for i=1:size(p_,2)
        p(:,i)=p_(1:2,i)/p_(3,i);
    end

    %Cálculo teniendo en cuenta la distorsión
    if distort==1

        %Paso de las coordenadas calculadas anteriormente (en píxeles)
        %a coordenadas en m, y normalizar.
        pn=[1/f*rox 0;0 1/f*roy]*(p-[u0;v0]);

        %Cálculo de posiciones distorsionadas
        for i=1:size(pn,2)
            pd(:,i)=pn(:,i)*(1+kr1*norm(pn(:,i))^2+kr2*norm(pn(:,i))^4)+...
                +[2*kt1*pn(1,i)*pn(2,i)+kt2*(norm(pn(:,i))^2+2*pn(1,i)^2);
                2*kt2*pn(1,i)*pn(2,i)+kt1*(norm(pn(:,i))^2+2*pn(2,i)^2)];
        end

        %Se deshace el paso realizado previamente a la aplicación de la
        %distorsión
        p=[f/rox 0;0 f/roy]*pd+[u0;v0];
    end
end

```

## %BLOQUE 3: REPRESENTACIÓN GRÁFICA

```

%Representación de la matriz de puntos en el espacio tridimensional
figure(j);subplot(1,2,1);
plot3(P(1,:),P(2,:),P(3,:), 'k. ');
grid;
xlabel('Eje X (m)');ylabel('Eje Y (m)');zlabel('Eje Z (m)');
title('Espacio real');

%Representación de los ejes {W} en el plano tridimensional
%(rojo:x, verde:y, azul:z)
ejewx=[wx wx+0.1;wy wy;wz wz];
ejewy=[wx wx;wy wy+0.1;wz wz];
ejewz=[wx wx;wy wy;wz wz+0.1];
line(ejewx(1,:),ejewx(2,:),ejewx(3,:), 'color','r','LineWidth',2);
line(ejewy(1,:),ejewy(2,:),ejewy(3,:), 'color','g','LineWidth',2);
line(ejewz(1,:),ejewz(2,:),ejewz(3,:), 'color','b','LineWidth',2);

%Representación de los ejes {C} en el plano tridimensional
%(rojo:x, verde:y, azul:z).
ejecx=WTC(:, :, j)*[ejewx;1 1];
ejecy=WTC(:, :, j)*[ejewy;1 1];

```

```

ejezc=WTC(:, :, j)*[ejewz;1 1];
line(ejecx(1,:),ejecx(2,:),ejecx(3,:), 'color','r','LineWidth',2);
line(ejecy(1,:),ejecy(2,:),ejecy(3,:), 'color','g','LineWidth',2);
line(ejezc(1,:),ejezc(2,:),ejezc(3,:), 'color','b','LineWidth',2);

%Representación del plano de la imagen
subplot(1,2,2);
%Marco del plano imagen
axis([-100 4228 -100 3196]);
rectangle('Position',[0 0 N M]);
hold on;
%Puntos proyectados
plot(p(1,:),p(2,:),'.');
%Proyección de los ejes Wx y Wy
line([p(1,1) p(1,(num_px-1)*num_py+1)], [p(2,1) p(2,(num_px-
1)*num_py+1)], 'color','r','LineWidth',1);
line([p(1,1) p(1,num_py)], [p(2,1) p(2,num_py)], 'color','g','LineWidth',1);
set(gca, 'YDir', 'reverse');
xlabel('Eje horizontal (pix)');ylabel('Eje vertical (pix)');
title('Plano imagen');
grid;

end

```