PABLO REDONDO

ALGORITMO PARA CAMBIAR DE COORDENADAS U-S-D Y HACER ZOOM DINAMICO CON LA RUEDA DEL RATÓN EN UN DRAWING AREA

Objetivo

Nuestro objetivo es hacer que al girar en un sentido o en otro la rueda del ratón sobre un drawing area, la imagen haga zoom tomando como centro del zoom la posición del cursor en ese momento.

Uno de los comandos más útil de los programas gráficos es la posibilidad de hacer zoom dinámico sólo con girar la rueda del ratón. También se utiliza a veces el zoom todo, pero mucho menos. El resto de los zooms se emplean muy raramente.

Fundamento teórico

Para dibujar cualquier cosa (punto, recta, arco, etc.) dentro de un drawing area necesitamos un sistema de coordenadas asociado a ese drawing area, y que llamaremos convencionalmente sistema de coordenadas screen, o simplemente S. La esquina superior izquierda del rectángulo del drawing area tendrá de coordenadas 0,0. El sentido positivo del eje x es hacia la derecha y, atención, el sentido positivo del eje y es hacia abajo. La unidad que he elegido para este sistema de coordenadas S será el metro.

Los objetos de nuestro mundo cotidiano los solemos referenciar con otro sistema de coordenadas, que llamaremos convencionalmente universal o U, en el que sentido positivo del eje x es hacia la derecha y, el sentido positivo del eje y es hacia arriba. La unidad que he elegido para este sistema de coordenadas es también el metro.

Tenemos un tercer sistema de coordenadas, que llamaremos dot o D, pues su unidad son los píxeles de la pantalla. El origen de coordenadas del sistema D y la orientación de sus vectores unitarios son los mismos que en el sistema S. Un punto de coordenadas 5,12 estará alejado 5 píxeles a la derecha del origen y 12 píxeles hacia abajo.

Entre los dos sistemas D y S podemos establecer una transformación de metros a píxeles, que aunque se expresan con unidades distintas, sus cantidades se corresponderán según la siguiente escala:

$$\text{escala} = \frac{D}{S}$$

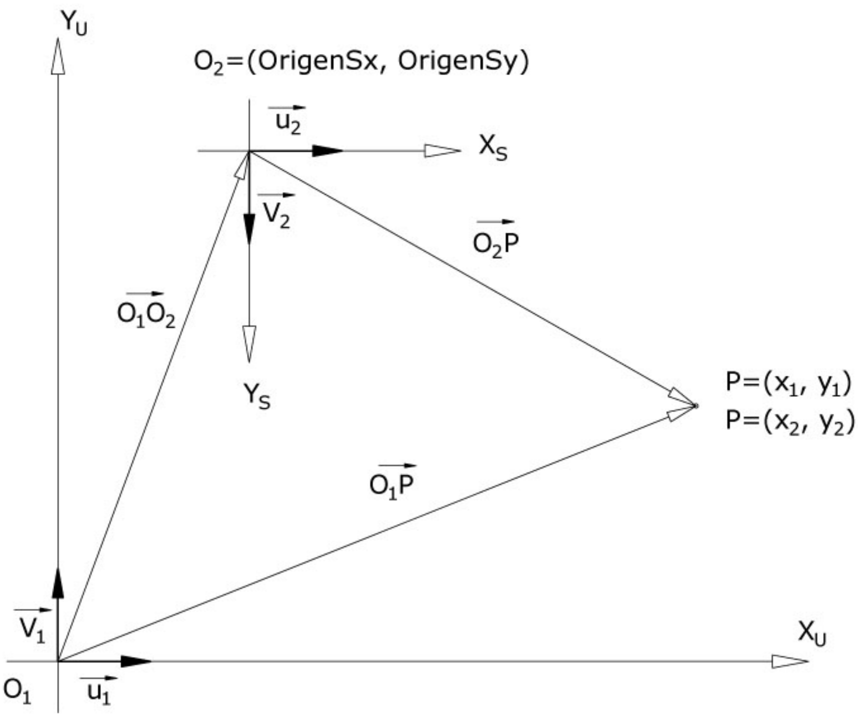
Para la pantalla de mi ordenador la escala = 4000, o si se prefiere escala = $\frac{4000}{1}$, es decir, que 4000 píxeles puestos todos seguidos es igual a 1 metro. Para que en la pantalla aparezca ese metro a escala 1/100 necesitaríamos hacer escala = 40, para que aparezca ese metro a escala 1/50 necesitaríamos hacer escala = 80, etc. Es decir, que una variable llamada escala, y que en principio puede valer 40, y que después podemos variar, nos relaciona los píxeles del sistema D con los metros del sistema S:

$$D = escala * S$$

$$S = \frac{D}{escala}$$

Volveremos más tarde sobre el tema de la escala, pero de momento vamos a imaginar que no existe, para poder tratar otro tema que es el cambio de coordenadas entre S y U.

Llamemos O_1 al origen del sistema U y $\vec{u}_1$ y $\vec{v}_1$ sus vectores unitarios. Llamemos O_2 al origen del sistema S y $\vec{u}_2$ y $\vec{v}_2$ sus vectores unitarios. El sistema S tendrá un origen O_2 respecto al sistema U, de coordenadas O_2=(OrigenSx, OrigenSy). Un punto cualquiera P tendrá unas coordenadas respecto al sistema U, $P=(x_1,y_1)$ y otras respecto al sistema S, $P=(x_2,y_2)$.



Los vectores unitarios pueden relacionarse mediante las siguientes igualdades:

$$\vec{u}_2 = \vec{u}_1 \qquad (1)$$

$$\vec{v}_2 = -\vec{v}_1 \qquad (2)$$

El cambio de coordenadas del sistema universal al sistema screen lo realizamos del siguiente modo:

$$\overrightarrow{O_1 O_2} + \overrightarrow{O_2 P} = \overrightarrow{O_1 P}$$

$$\overrightarrow{O_2 O_P} = \overrightarrow{O_1 P} - \overrightarrow{O_1 O_2}$$

$$x_2 \vec{u_2} + y_2 \vec{v_2} = x_1 \vec{u_1} + y_1 \vec{v_1} - (OrigenSx \cdot \vec{u_1} + OrigenSy \cdot \vec{v_1})$$

Igualando las coordenadas x con las x, y las coordenadas y con las y, tenemos dos igualdades:

$$x_2 \cdot \vec{u_2} = (x_1 - OrigenSx) \cdot \vec{u_1}$$

y como (1) podemos deducir que:

$$x_2 = x_1 - OrigenSx \qquad (3)$$

Y la segunda igualdad:

$$y_2 \cdot \vec{v_2} = (y_1 - OrigenSy) \cdot \vec{v_1}$$

y teniendo en cuenta (2):

$$y_2 \cdot \vec{v_2} = (y_1 - OrigenSy) \cdot \vec{v_1} = (OrigenSy - y_1) \cdot \vec{v_2}$$

$$y_2 = OrigenSy - y_1 \qquad (4)$$

Si quisiéramos en cambio trasformar las coordenadas de S en las coordenadas de U, podemos despejar directamente de (3) y (4):

$$x_1 = x_2 + OrigenSx \qquad (5)$$

$$y_1 = OrigenSy - y_2 \qquad (6)$$

Ahora vamos a juntarlo todo, la transformación de la escala con la transformación de coordenadas, para tener la transformación total entre un sistema U y un sistema D. Vamos a transformar de U $\rightarrow$ D, P=(x_1,y_1) $\rightarrow$ P=(x_3,y_3):

1.- Transformamos las coordenadas con (3) y (4):

$$x_2 = x_1 - OrigenSx \qquad (3)$$

$$y_2 = OrigenSy - y_1 \qquad (4)$$

2.- Transformamos los metros a píxeles con la escala S $\rightarrow$ D:

$$x_3 = x_2 \cdot escala \qquad (7)$$

$$y_3 = y_2 \cdot escala \qquad (8)$$

Todo ello junto lo podemos sintetizar en sólo paso:

U $\rightarrow$ D, P=(x_1,y_1) $\rightarrow$ P=(x_3,y_3):

$$x_3=(x_1-OrigenSx)\cdot escala \qquad (9)$$

$$y_3=(OrigenSy-y_1)\cdot escala \qquad (10)$$

Y al contrario, los pasos para tener la transformación total entre un sistema D y un sistema U. D $\rightarrow$ U, P=(x_3,y_3) $\rightarrow$ P=(x_1,y_1):

1.- Transformamos los píxeles a metros con la escala:

$$x_2=x_3/escala \qquad (11)$$

$$y_2=y_3/escala \qquad (12)$$

2.- Transformamos las coordenadas con (5) y (6):

$$x_1=x_2+OrigenSx \qquad (5)$$

$$y_1=OrigenSy-y_2 \qquad (6)$$

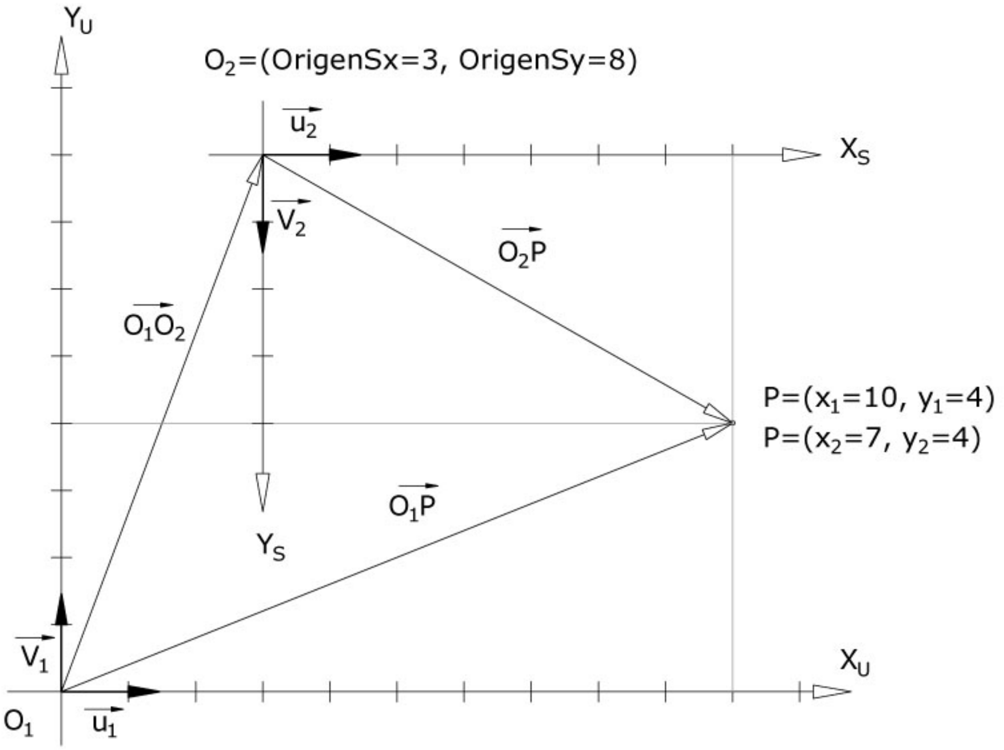
Y de nuevo, todo ello junto lo podemos sintetizar en sólo paso:

D $\rightarrow$ U, P=(x_3,y_3) $\rightarrow$ P=(x_1,y_1):

$$x_1=x_3/escala+OrigenSx \qquad (13)$$

$$y_1=OrigenSy-y_3/escala \qquad (14)$$

En unas ocasiones nos interesará pasar U $\rightarrow$ D (datos: x_1, y_1, OrigenSx, OrigenSy) y en otras D $\rightarrow$ U (datos: x_3, y_3, OrigenSx, OrigenSy). Hay que recordar que las coordenadas del origen O_2=(OrigenSx, OrigenSy) son siempre respecto al sistema U, son siempre coordenadas universales, en metros, puesto que respecto al sistema S o al D serán siempre cero O_2=O_3=(0, 0).



Pongamos un ejemplo práctico, en principio sin considerar la transformación por escala:

U $\rightarrow$ S, P=(x_1,y_1) $\rightarrow$ P=(x_2,y_2):

$$x_2 = x_1 - OrigenSx$$
$$x_2 \ = 10\text{-}3 = 7$$
$$y_2 = OrigenSy - y_1$$
$$y_2 \ = 8\text{-}4 = 4$$

S $\rightarrow$ U, P=(x_2,y_2) $\rightarrow$ P=(x_1,y_1):

$$x_1 = x_2 + OrigenSx$$
$$x_1 \ = 7+3 = 10$$

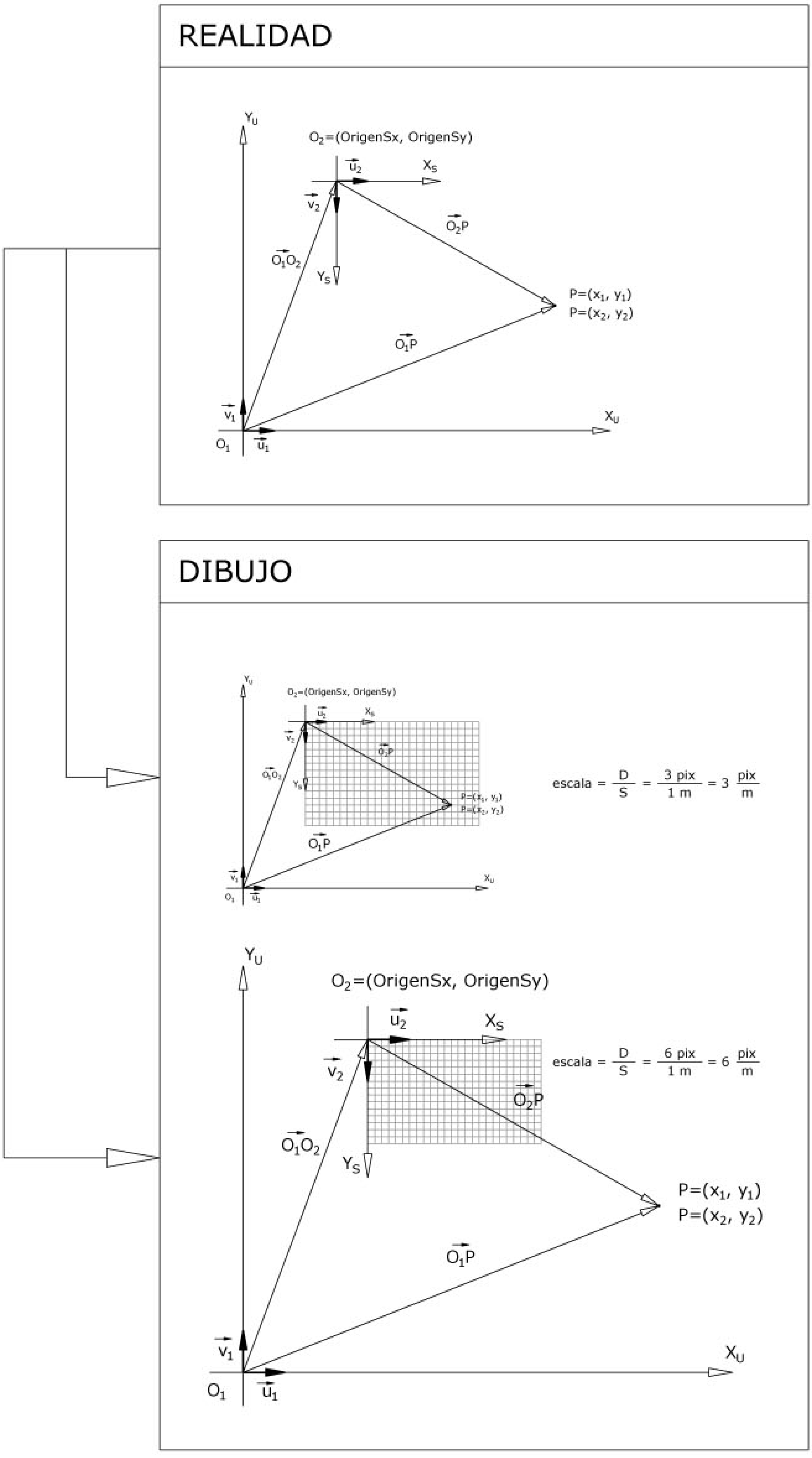
$$y_1 = OrigenSy - y_2$$
$$y_1 \ = 8\text{-}4 = 4$$

Hasta aquí sólo hemos explicado cómo vamos a cambiar del sistema Universal U al de la Dots D y a la inversa. Vamos a describir ahora el fundamento teórico para hacer zoom dinámico.

Un zoom consiste en hacer dos trasformaciones: cambiar el valor de la escala que hemos explicado anteriormente, la escala que nos relaciona píxeles con metros, D con S, y también cambiar el origen del sistema S (el mismo origen siempre que el origen del sistema D) cuyas coordenadas son (OrigenSx, OrigenSy) respecto al sistema de coordenadas universal U.

La trasformación de la escala "sólo" coge la realidad y la amplía o reduce, con lo que el resultado es nuestro dibujo. Si ponemos menos píxeles por cada metro, nos cabrán más metros en el drawing area y la realidad será sometida a una reducción. Y al contrario, si ponemos más píxeles por cada metro, nos cabrán menos metros en el drawing area y la realidad será sometida a una ampliación. En la siguiente imagen la realidad ha sido reducida a dos escalas diferentes, a 3 píxeles por metro y a 6 píxeles por metro. Y decimos que la trasformación de la escala "sólo" coge la realidad y la amplía o reduce porque en ambos casos el origen del sistema S no ha cambia, sus coordenadas siguen siendo (OrigenSx, OrigenSy) respecto al sistema de coordenadas universal U:

TRANSFORMACION DE METROS A PÍXELES



Cuando queremos hacer zoom dinámico, el cursor se encuentra en un píxel determinado de nuestra drawing area. Lo primero que vamos a hacer es averiguar las coordenadas universales de ese píxel, para lo cual utilizamos las fórmulas (13) y (14) que hemos visto antes.

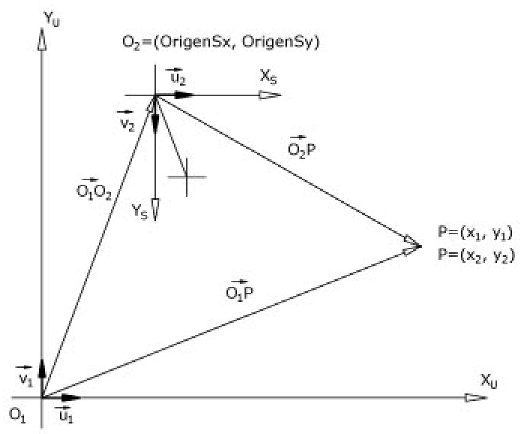
D $\rightarrow$ U, P=(x_3,y_3) $\rightarrow$ P=(x_1,y_1):

$$x_1 = x_3 / escala + OrigenSx \qquad (13)$$

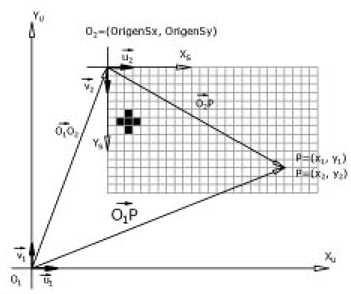
$$y_1 = OrigenSy - y_3 / escala \qquad (14)$$

El punto así obtenido está separado una distancia del origen del sistema S. Lo vemos en el siguiente dibujo. Ahora ya no tenemos un píxel sino un punto de coordenadas (Ratonx, Ratony) respecto al sistema de coordenadas universal U:

REALIDAD



DIBUJO



$$\text{escala} = \frac{D}{S} = \frac{3\ \text{pix}}{1\ \text{m}} = 3\ \frac{\text{pix}}{\text{m}}$$

Ahora lo que necesitamos es definir el Incremento de zoom que ocasionará un paso de rueda de ratón. Supongamos que nos queremos "acercar" a nuestro dibujo, es decir, que ahora la escala no será de 3 píxeles por metro sino el doble, 6 píxeles por metro. El Incremento de escala será por lo tanto 2. Esto quiere decir que vamos a acercar el origen del sistema de coordenadas S al punto sobre el que se encuentra el ratón. Hacemos:

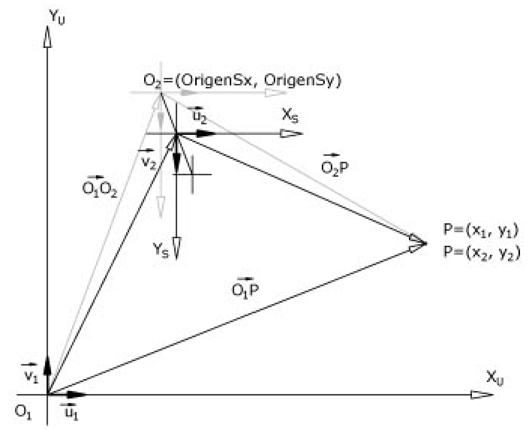
OrigenSx = OrigenSx + (Ratonx - OrigenSx) * (Incremento - 1) / incremento
OrigenSy = OrigenSy + (Ratony - OrigenSy) * (Incremento - 1) / incremento

Si en lugar de acercarnos a nuestro dibujo (acercar el origen de S) quisiéramos alejarnos del dibujo (alejar el origen de S), las fórmulas son:

OrigenSx = OrigenSx + (OrigenSx - Ratonx) * (Incremento - 1)
OrigenSy = OrigenSy + (OrigenSy - Ratony) * (Incremento – 1)

Las coordenadas P (x_1, y_1) de un punto cualquiera punto P son siempre las mismas, pero al cambiar el origen del sistema S, las coordenadas P (x_2, y_2) respecto a este sistema habrán cambiado:

REALIDAD



Y_U
O_2=(OrigenSx, OrigenSy)
$\vec{u_2}$
X_S
$\vec{v_2}$
$\overrightarrow{O_2P}$
$\overrightarrow{O_1O_2}$
Y_S
P=(x_1, y_1)
P=(x_2, y_2)
$\overrightarrow{O_1P}$
$\vec{v_1}$
X_U
O_1
$\vec{u_1}$

Una vez que hemos cambiado el origen del sistema S, lo único que queda para completar el zoom es redibujar todos los elementos del sistema aplicando la nueva escala de transformación.

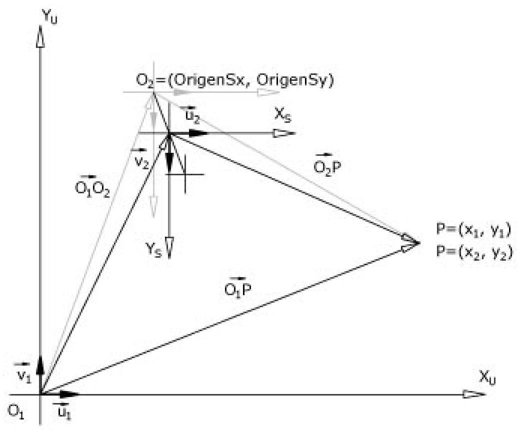
Si nos acercamos:

escala = escala * Incremento

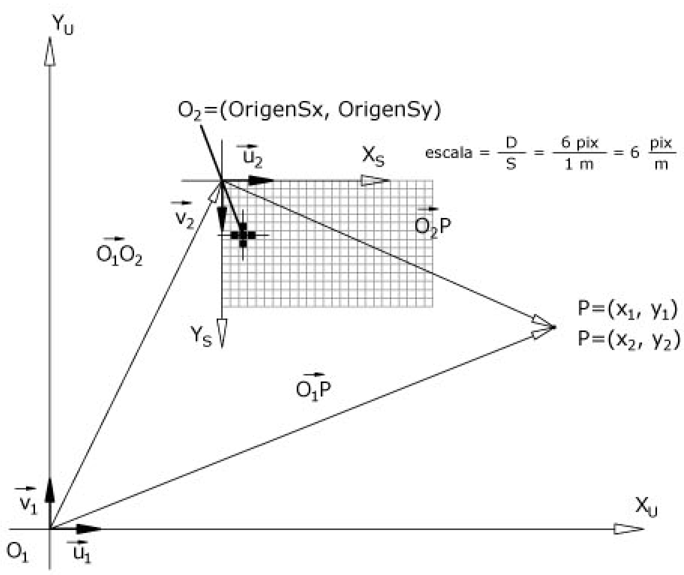
Si nos alejamos:

escala = escala / Incremento

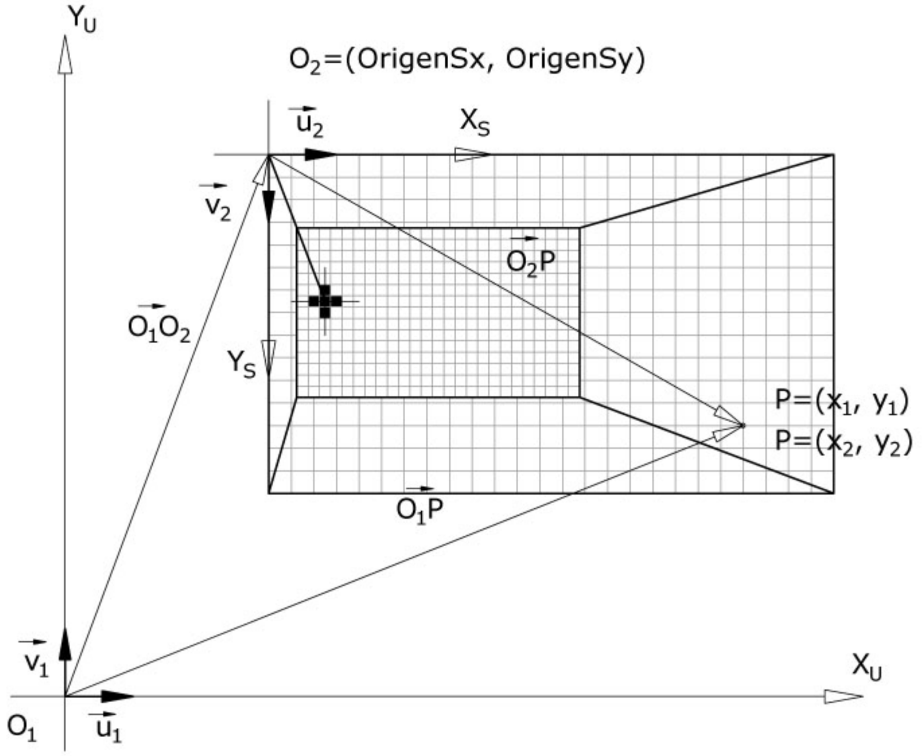
REALIDAD



DIBUJO



Con esto hemos conseguido que el zoom dinámico haga escala tomando como centro el punto sobre el que se encontraba el ratón. El siguiente dibujo mantiene constante la realidad (en lugar de ampliarla o reducirla, que sería lo propio) y lo que hace es escalar el tamaño de los píxeles, con lo cual podemos ver más intuitivamente qué es lo que ha ocurrido:



El cuadro del drawing area crece o decrece con centro en el cursor, cogiendo más o menos campo de la realidad. El número de píxeles es el mismo, sólo que en esta explicación intuitiva los hemos hecho más o menos gordos.

La subrutina de zoom dinámico se activará dentro de un evento del drawing area. Si, por ejemplo, llamamos al drawing area con el nombre da, la subrutina de evento será da_MouseWheel(). Esta subrutina lo único que va a hacer es cambiar el origen del sistema S (OrigenSx, OrigenSy) y la escala, según el valor que hayamos definido para el Incremento. Un valor de Incremento = 1.4 no está mal, pero es cuestión de hacer pruebas, o incluso dar al usuario la posibilidad de ajustar ese parámetro a su equipo.

En esto consiste el zoom dinámico que queríamos.

Pablo Redondo
pabloredondoarq@gmail.com

Octubre 2012