Estado del Arte, Herramientas y Aplicaciones para Transformaciones geométricas 3D

Diego Fernando Villamarín Zapata
Universidad de las Fuerzas Armadas - ESPE
Quito - Ecuador
E-mail: dfvillamarin@espe.edu.ec

Resumen-- El presente trabajo muestra una breve revisión del estado del arte de las transformaciones de la geometría en 3D, detalla varios métodos, herramientas existentes y aplicaciones interesantes. El propósito fundamental es dar una visión general para quienes estén interesados en empezar a explorar las transformaciones geométricas en objetos 3D, estudiando las diferentes transformaciones existentes para generar movimientos, rotaciones, traslaciones, escalamiento, proyecciones, deformaciones, etc. Además se presenta varias herramientas disponibles para desarrollar aplicaciones con transformaciones geométricas. Finalmente se presenta varios ejemplos de aplicaciones reales que se desarrollan en este campo y su importancia en la sociedad.

Abstract--This paper presents a brief review of related literature in 3D geometry transformations, detailed methods, existing tools and interesting applications in this setting. The main purpose is to provide an overview for people interested in starting to explore geometric transformations of 3D objects, and studying some transformations to generate movements, rotations, translations, scaling, projections, deformations, etc. Additionally, some available tools are presented for developing applications with geometric transformations. Finally, several examples of real applications being developed in this field and the importance that they have in the society are presented.

Palabras claves-- **Transformaciones Geométricas, 3D, Transformación Afín, Transformación Perspectiva.**

I. INTRODUCCIÓN

El conjunto de operaciones elementales que realiza internamente un computador para conseguir pasar de la representación de un modelo geométrico tridimensional a su imagen en pantalla 2D, pero que para la impresión del observador parezca estar contemplando un sistema de visualización en el mundo real con tres dimensiones 3D, utiliza varias de las transformadas geométricas que se describirán en las siguientes secciones.

Estos modelos geométricos tridimensionales deben someterse a ciertas transformaciones antes de que su imagen aparezca en la pantalla de un dispositivo. Existen varias primitivas geométricas dadas por los modeladores 3D: esferas, cilindros, toros, parches paramétricos, curvas racionales diferenciables definidas por polinomios no uniformes conocidas como NURBS, etc., sin embargo, a más bajo nivel, todos los algoritmos que se utilizan se basan en una única primitiva: el polígono. Internamente los polígonos se dividen en elementos más simples: triángulos (única primitiva geométrica que puede manejar un acelerador gráfico) [4].

El presente trabajo está organizado como se describe a continuación. En la sección II, se detallan, los antecedentes, conceptos básicos y generalidades sobre las transformadas geométricas; la sección III muestra los tipos de transformadas afín; la sección IV muestra la transformación perspectiva y bilineal; la sección V muestra algunas herramientas disponibles para utilizar estas transformadas; en la sección VI se muestran varios ejemplos de aplicaciones, finalmente en la sección VII, se incluye conclusiones.

II. TRANSFORMACIONES DE LA GEOMETRÍA

Las transformaciones geométricas se definen como la relaciones de los puntos en dos imágenes, se representan como operaciones matriciales sobre los puntos del objeto y cada uno se representa como una matriz constituida por las coordenadas (x, y, z) de los puntos que forman dicho objeto.

Como algunas transformaciones se obtienen por multiplicación de matrices y otras por sumas de vectores (como por ejemplo la traslación) se utiliza un truco matemático: representar el término independiente como una cuarta coordenada extra. A esa cuarta coordenada se le denomina coordenada homogénea. En los puntos, la cuarta coordenada es un 1, en los vectores la cuarta coordenada es un 0 [4].

La matriz de transformación en coordenadas homogéneas de acuerdo a los diferentes tipos de operaciones con elementos geométricos se muestra en la figura 1.

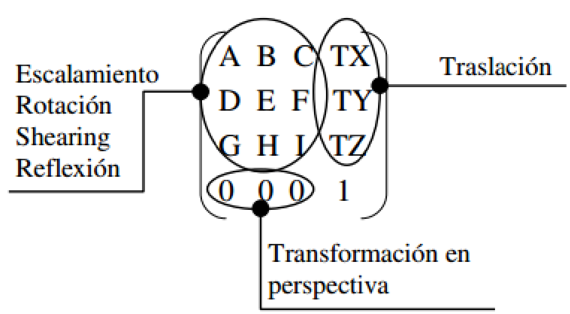

Fig. 1. Matriz de transformación en coordenadas homogéneas.

Escrito el 05 de Febrero de 2015. Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas.

Se considera como transformaciones básicas o elementales a la traslación, la rotación, el cambio de escala, a la simetría y a la deformación o shearing.

De la figura 1, el bloque 3x3 produce una transformación lineal que provoca dilataciones o escalamientos, deformaciones, reflexiones y rotaciones. El bloque de la fila 1x3 genera una transformación de la perspectiva y la columna 3x1 produce una translación, el elemento del bloque 1x1 actúa como factor de escala que dilata y contrae globalmente el cuerpo [5].

III. Transformación Afín

Las transformaciones afines son aquellas que conservan la rectitud y el paralelismo, además mantiene las proporciones a lo largo de las rectas. Una transformada afín es una transformación lineal de una coordenada, que incluye las transformadas elementales, traslación, rotación, escalado y deformación o inclinación como se muestra en la figura 2 [2].

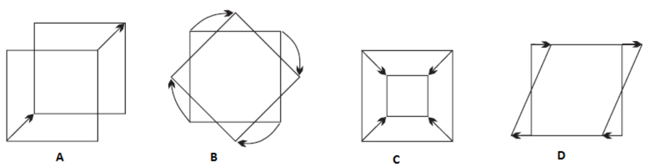

Fig. 2. A: Traslación. B: Rotación, C: Escalado y D: Deformación.

Las transformaciones afines solamente el escalamiento modifica el tamaño, y solo hay un caso en el que existe conmutatividad: rotación y cambio de escala uniforme, salvo esa situación, no existe la conmutatividad, por lo que el orden de aplicación de una de ellas es muy importante.

A. Translación en 3D

Se llama traslación en el espacio 3D al desplazamiento de un poliedro, donde cada punto $p=(X_1,X_2,X_3)$ es trasladado d_1 unidades en el eje X_1, d_2 unidades en el eje X_2 y d_3 unidades en el eje X_3, de esta forma, las coordenadas del punto $p'=(X_1',X_2',X_3')$ se obtienen como:

$$X_1'=X_1+d_1$$
$$X_2'=X_2+d_2$$
$$X_3'=X_3+d_3$$

Sea $d=(d_1,d_2,d_3)$ el vector de distancias, y $T(d)$ la matriz de traslación, en coordenadas homogéneas la traslación de un punto p en 3D se puede expresar como el producto matricial $p'=p.T(d)$, es decir: [6].

$$[X_1'\ X_2'\ X_3'\ 1]=[X_1\ X_2\ X_3\ 1].\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_1 & d_2 & d_3 & 1 \end{bmatrix}$$

Ecuación 1. Expresión Matricial para la translación 3D.

B. Rotación en 3D

A diferencia de la rotación en el espacio 2D, donde para hacer rotar un objeto se necesita un punto (cero-dimencional), en 3D para hacer rotar un objeto se necesita dos puntos no coincidentes que determinan un segmento de recta, cuya línea de soporte define un eje lineal (unidimensional) de rotación.

Las rotaciones 3D principales, son aquellas cuando el eje de rotación se encuentra sobre alguno de los tres ejes principales: X_1,X_2 o X_3, las rotaciones sobre cualquier otro eje arbitrario son llamadas rotaciones generales 3D.

Para entender el concepto de rotación en 3D como una extensión de la rotación 2D, hay que saber que la rotación 2D es el giro sobre el eje de rotación, que es perpendicular al plano X_1,X_2, el cual en 3D pertenece al eje X_3, entonces se tiene la primera de las rotaciones principales.

De esta forma para cada punto $p=(X_1,X_2,X_3)$, dado un ángulo θ, puede ser rotado sobre el eje X_3, en sentido contrario de las manecillas del reloj, obteniendo las coordenadas en el punto $p'=(X_1',X_2',X_3')$, quedando la coordenada X_3 sin cambio, entonces, se extiende las fórmulas para la rotación 3D como:

$$\begin{cases} X_1'=X_1\cos\theta-X_2\sin\theta \\ X_2'=X_1\sin\theta-X_2\cos\theta \\ X_3'=X_3 \end{cases}$$

Sea $R_3(\theta)$ la matriz de rotación sobre el eje X_3, en coordenadas homogéneas la rotación de un punto p alrededor de dicho eje, se puede expresar como el producto matricial $p'=p.R_3(\theta)$, es decir: [6].

$$[X_1'\ X_2'\ X_3'\ 1]=[X_1\ X_2\ X_3\ 1].\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 2. Expresión matricial de la rotación 3D alrededor del eje X3.

C. Escalamiento en 3D

El escalamiento o escalado 3D implica el cambio de tamaño de un poliedro, donde cada punto $p=(X_1,X_2,X_3)$ es transformado por la multiplicación de tres factores de escalamiento: S_1,S_2 y S_3 a lo largo de los ejes X_1,X_2 y X_3 respectivamente, de esta forma las coordenadas del punto $p'=(X_1',X_2',X_3')$ se obtienen como:

$$X_1'=X_1.S_1$$
$$X_2'=X_2.S_2$$
$$X_3'=X_3.S_3$$

Sea $S=(S_1,S_2,S_3)$ el vector de factores de escalamiento, y $S(s)$ la matriz de escalonamiento, en coordenadas homogéneas el escalonamiento de un punto p en 3D se puede expresar como el producto matricial $p' = p.S(s)$, es decir:

$$[X_1'\ X_2'\ X_3'\ 1] = [X_1\ X_2\ X_3\ 1].\begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3. Expresión Matricial para el escalamiento en 3D.

D. Deformación en 3D

Una transformación de deformación o de inclinación en la dirección del eje X y que conserve el plano horizontal tiene la forma $(x, y, z) = (x + cz, y, z)$ en donde r es el factor de deformación. La matriz de transformación en coordenadas homogéneas es:

$$\begin{bmatrix} 1 & 0 & c & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los términos no diagonales del bloque 3x3 de la matriz de transformación producen deformaciones de la siguiente forma:

$$(x\ y\ z\ 1).\begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ h & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (x + yd + hz \quad bx + y + iz \quad cx + fy + z \quad 1)$$

Ecuación 4. Expresión Matricial para la deformación en 3D.

El elemento (i, j) de la matriz T produce una deformación de la coordenada X_j en función de la X_i, siendo $x_1 = x, \quad x_2 = y, \quad x_3 = z.$ Así por ejemplo;

$$(x\ y\ z\ 1).\begin{bmatrix} 1 & b & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (x \quad bx + y \quad z \quad 1)$$

En el ejemplo anterior se puede ver que la coordenada y de (x y z) varía linealmente con x por efecto del término (1,2).

IV. Transformación Perspectiva y Bilineal

Las transformaciones perspectivas o también conocidas como proyectivas, son aquellas que transforman líneas rectas en líneas rectas, pero que no necesariamente conservan el paralelismo [5].

La transformada bilineal se suele usar como una variante rápida de la transformada perspectiva, aunque no es exactamente igual. Es decir, se debería aplicar perspectiva, pero se usa la bilineal por eficiencia. La transformada bilineal es una simulación de la perspectiva. También mapea un rectángulo en un cuadrilátero, pero el resultado no es exactamente una perspectiva y la diferencia es mayor cuanto mayor efecto de perspectiva exista [8].

En contraste con las transformadas afines, las transformaciones proyectivas no son lineales, sin embargo, esta transformada es reducida a una transformada lineal utilizando coordenadas homogéneas [2].

Toda transformación proyectiva espacial puede ser expresada en coordenadas homogéneas mediante una matriz 4x4 invertible. Recíprocamente, toda matriz de 4x4 invertible define una transformación proyectiva sobre el plano [5].

Las transformaciones bilineal y perspectiva se pueden ver como generalizaciones de las afines:

- Transformación afín: cualquier rombo se mapea en un rombo (ver figura 3.)
- Transformación bilineal y perspectiva: cualquier cuadrilátero se transforma en otro cuadrilátero (ambos convexos), como se muestra en la figura 3.

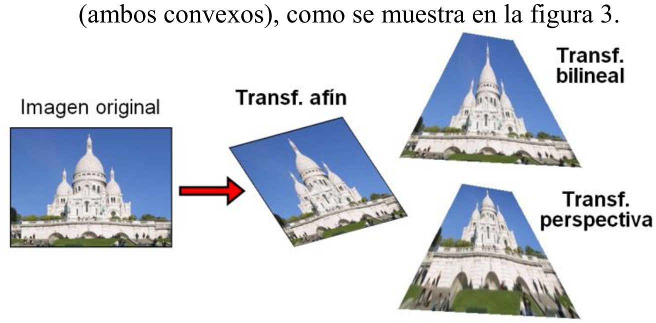

Fig. 3. Diferencias entre transformada afín, bilineal y perspectiva.

La representación de una figura del espacio sobre un plano es una aplicación habitual de las transformaciones proyectivas, de forma que para realizar la proyección de una figura del espacio sobre la pantalla del ordenador es necesario realizar una trasformación proyectiva.

Otra aplicación habitual es la denomina calibración de la cámara, se le conoce así a la determinación de la correspondencia entre puntos de la imagen y puntos en el mundo real. Consiste en determinar los valores de los parámetros del modelo de cámara empleado para una cámara particular, de manera que pueda modelarse su funcionamiento desde el punto de vista geométrico. Dada una plantilla con numerosos puntos distinguibles, de la que se conozcan sus posiciones 2D y 3D, estos se sustituyen en las ecuaciones del modelo, siendo las incógnitas los parámetros de la cámara; cuantos más puntos se usen más precisa será la calibración. La transformación perspectiva directa se utilizaría para pasar del mundo 3D al 2D y la transformación perspectiva inversa para pasar de 2D a 3D [7].

V. Herramientas disponibles

Una vez que se ha revisado las principales operaciones con las transformaciones geométricas que permiten modificar la posición y orientación de los objetos, es importante conocer herramientas para poder aplicarlas, una de las más utilizadas es **OpenGL** (Open Graphics Library) esta es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más

ISSN: 1390-4671

de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en diseño asistido por computador CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compite con Direct3D [10].

No hace falta ser un experto matemático para poder hacer uso de las funciones que ofrece OpenGL para realizar transformaciones geométricas. Lo que si se necesita, es conocer los aspectos básicos que permitan saber qué cosas se pueden hacer y qué herramientas serán las mejores para conseguir nuestros propósitos. Cuando se habla de gráficos 3D realmente no estamos hablando de objetos en 3D, sino que se trata conceptos 3D para describir cómo un objeto 3D puede representarse en un monitor 2D. A este proceso se le denomina “proyección”. Por tanto los objetos 3D se proyectan en el plano, pero hay varias formas de ver esta proyección, la proyección en perspectiva y ortogonal. En la primera los objetos más lejanos se verían más pequeños, que es lo que en realidad sucede, mientras que en la segunda todos los objetos se ven a la misma distancia y tienen el mismo tamaño [9].

La proyección es una de las transformaciones que permite utilizar OpenGL pero existen muchas más. Las transformaciones nos permitirán por ejemplo rotar objetos, desplazarlos, escalarlos, etc. Un ejemplo de ello se muestra en la figura 4.

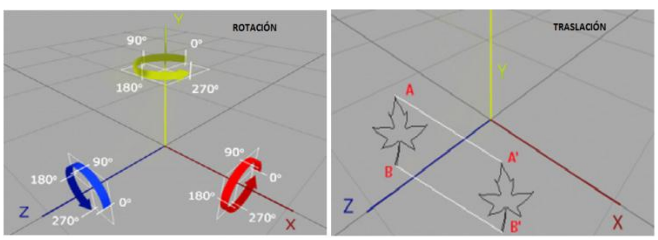

Fig. 4. Ejemplo de rotación y traslación de objetos sobre los ejes de coordenadas X, Y, Z, con OpenGL.

Es muy importante hacer hincapié en que las transformaciones geométricas se aplican antes de que un objeto sea visualizado. A continuación se muestra algunos códigos de ejemplo OpenGL.

Traslación: Si se quiere pintar un cubo de 10 unidades y desplazarlo 10 unidades sobre el eje X. El código sería:

```
// Desplaza 10 unidades sobre el eje X
glTranslatef(10.0f, 0.0f, 0.0f);
//Pinta el cubo
glutWireCube(10.0f);
```

Rotación: Para rotar un objeto sobre uno de los 3 ejes de coordenadas, o sobre cualquier otro vector definido V (x,y,z), OpenGL nos permite utilizar la función:

```
glRotatef (GLfloat angulo, GLfloat x, GLfloat y, GLfloat z);
```

El ángulo de rotación es siempre un ángulo en sentido en contra de las agujas del reloj y medido en grados. Si por ejemplo quisiéramos rotar 45 grados nuestro cubo sobre el eje x el código sería el siguiente:

```
// Realiza la rotación
glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
```

Escalado: es una transformación que permite cambiar el tamaño de un objeto expandiendo o contrayendo todos sus vértices. La función que nos permite realizar el escalado en OpenGL es la siguiente:

```
glScalef (GLfloat x, GLfloat y, GLfloat z);
```

El escalado no tiene porqué ser uniforme, y podemos expandir por tanto un objeto más en anchura que en altura. Un ejemplo que ensancha el tamaño del cubo al doble en el eje X:

```
// Realiza el escalado
glScalef( 2.0f, 1.0f, 1.0f );
```

Cuando se trata de aplicar una sola transformación no hay ningún problema con las funciones anteriormente detalladas, pero en el caso de que se quiera realizar varias transformaciones a un objeto se tendrá que entender mejor cómo las gestiona y lleva a cabo OpenGL. La idea principal es que OpenGL utiliza una pila (LIFO) para almacenar las transformaciones.

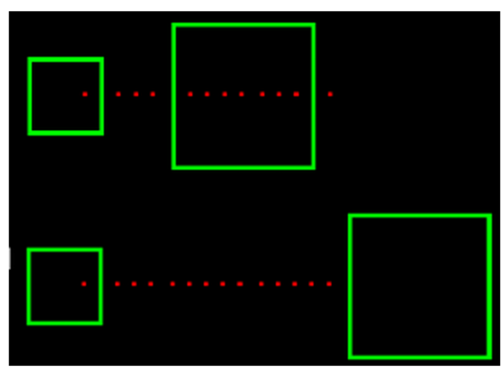

Fig. 5. Ejemplo de traslación y escalado de objetos, con OpenGL.

En la figura 5, se observa la diferencia entre ejecutar una traslación primero y un escalado en segundo lugar, y ejecutar primero un escalado y luego una traslación:

```
//primer ejemplo
glScalef( 2.0f, 2.0f, 2.0f);
glTranslatef( 10.0f, 0.0f, 0.0f);
glutWireCube(4.0f);

//segundo ejemplo
glTranslatef( 10.0f, 0.0f, 0.0f);
glScalef( 2.0f, 2.0f, 2.0f);
```


ISSN: 1390-4671

```
glutWireCube(4.0f);
```

Otra herramienta también de distribución abierta es P3D (Processing 3D), Processing es un lenguaje de programación y entorno de desarrollo para la comunidad en línea. Desde el año 2001, Processing ha promovido la inclusión de software dentro de las artes visuales. Inicialmente creado para servir como un cuaderno de bocetos por software y para enseñar los fundamentos de programación de computadoras dentro de un contexto visual, Processing evolucionó hasta convertirse en una herramienta de desarrollo para profesionales. Hoy en día, hay decenas de miles de estudiantes, artistas, diseñadores, investigadores y aficionados que la utilizan para el aprendizaje, la creación de prototipos y producción [11].

Algunas de las directivas para las transformaciones geométricas en 3D con processing:

- Traslación
  translate(tx, ty, tz)
- Escalado
  scale(sx, sy, sz)
- Rotación respecto a un eje
  rotateX(), rotateY(), rotateZ()

En la figura 6 se observa la compilación del siguiente código, como resultado se obtiene un cubo rotado alrededor de los tres ejes [12]:

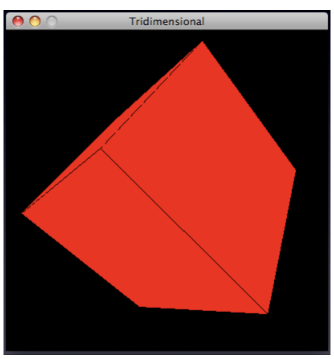

Fig. 6. Ejemplo de traslación y rotación de un objeto, con P3D.

```
// Versión sólida
void setup()
{
 size(400, 400, P3D);
 fill(255, 0, 0);
}
void draw()
{
 background(0);
 // Dibujo centrado en el (0,0,0)
 translate(width/2, height/2);
 rotateX(frameCount*PI/60.0);
 rotateY(frameCount*PI/120.0);
 rotateZ(frameCount*PI/180.0);
 box(200, 200, 200);
}
```

VI. Aplicaciones 3D con Transformaciones

En las imágenes 3D, existe información que son invariantes ante rotaciones como por ejemplo, las distancias, las áreas, los ángulos, etc. Y esta característica es útil para desarrollar varias aplicaciones como por ejemplo:

- Estimación de la forma precisa de objetos o de algunas de sus características geométricas, utilizada en inspección de objetos en entornos industriales.
- Reconocimiento de objetos a partir de sus propiedades tridimensionales:
  - Identificación tejidos celulares en Biología,
  - Obtención de la distancia de un robot a un objeto, la distancia entre dos objetos o las dimensiones de un objeto.
  - Manipulación de objetos en Robótica, detección de colisiones en navegación de robots autónomos, etc. [7].

Un ejemplo típico de las transformaciones afines es extraer y redimensionar un área de interés, dándole una forma predefinida de antemano. Esto es lo que se llama normalización. Por ejemplo, detectar una cara humana, seleccionar los ojos y la boca y mapearlos a un rectángulo predefinido como se muestra en la figura 7.

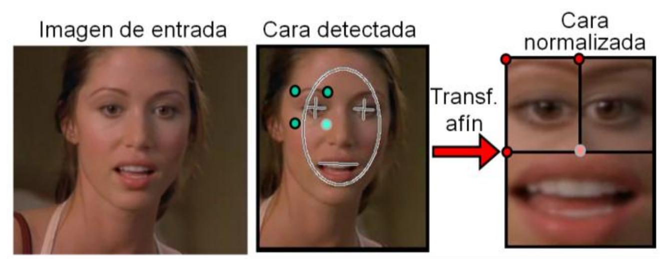

Fig. 7. Ejemplo de aplicación con transformaciones afín.

Este proceso de normalización se puede aplicar sobre vídeo, para conseguir una estabilización de los objetos de interés. La normalización es fundamental en muchas aplicaciones de reconocimiento de objetos, como los OCR (Optical Character Recognition) [8].

Siguiendo la aplicación en el área de reconocimiento facial, los resultados de un experimento muestran que el rendimiento del reconocimiento facial puede ser fuertemente comprometido por la transformación de la perspectiva en imágenes, independientemente de si los rostros se muestran en la vista de toda la cara o en los tres cuartos de la vista [13]. Para citar un ejemplo claro donde se utiliza la transformación bilineal y perspectiva se describe el proceso de la calibración o integración de elementos visuales artificiales en un entorno, el objetivo es hacer que algo que no está parezca que realmente si está, como se observa en la figura 8, donde sobre el campo de juego se integra el marcador con los sellos de los equipos y una publicidad. Para lograr esto primero se detecta el suelo en este caso de color verde, después se realiza la transformación perspectiva de cada uno de los elementos a integrar en el entorno y finalmente se realiza un proceso de ponderación entre el suelo y la imagen transformada.

Fig. 8. Ejemplo de aplicación con transformación perspectiva.

Para lograr esto primero se detecta el suelo en este caso de color verde, después se realiza la transformación perspectiva de cada uno de los elementos a integrar en el entorno y finalmente se realiza un proceso de ponderación entre el suelo y la imagen transformada.

En otro ejemplo de aplicación la transformación perspectiva se usa para proyectar puntos del horizonte delante de la cámara a un plano de imagen y por lo tanto para medir la distancia entre el objeto detectado y la cámara. Con la información de los puntos sobre el terreno, las posiciones proyectadas de cada punto del objeto rígido, pueden imaginarse través de la transformación, características de la proyección del objeto a distancias diferentes, tales como tamaño y forma, también se los puede predecir [14].

La transformación de la perspectiva basada en la geometría de la cámara estereoscópica es ampliamente utilizada en la visión artificial en 3D [15], y usando la transformación de perspectiva inversa se ha implementado una solución de bajo costo, simple y eficiente para aprovechar la navegación visual de robots móviles, a través de una simple cámara, que por medio de la detección de obstáculos pueden ser manejados online y en tiempo real [16].

VII. Conclusiones

En este trabajo se muestra una visión general de las transformadas geométricas aplicadas en 3D, se describe algunos de los tipos de transformaciones desde las más elementales como hacer una escala en una imagen hasta otras con mayor complejidad y se muestran algunos ejemplos típicos de aplicaciones.

Las transformaciones afines, bilineales y perspectivas son esenciales en generación, procesamiento, análisis de imágenes, y en visión artificial. Hay que conocer el significado de cada trasformación para saber cuál conviene aplicar y en qué orden aplicarla.

Finalmente se concluye que las transformaciones de la geometría en el mundo 3D, el desarrollo y distribución de sus aplicaciones especialmente aplicados en la visión artificial para la navegación de robots en tiempo real, el reconocimiento de imágenes aplicado en fotomultas y en procesos industriales como selección de alimentos o productos en buen estado, sin duda están en auge ya que cada vez estas aplicaciones se integran para facilitar nuestras actividades cotidianas brindando soluciones rápidas, eficientes, y que están al alcance de todos, gracias a varias herramientas de distribución gratuita disponibles para su desarrollo.

Referencias

[1] William K Pratt, Digital Image Processing, Fourth Edition, pp. 408, 2007.
[2] Bernd Jahne, Digital Image Processing, Sixth revised and extended Edition, pp. 275, 2005.
[3] R.C. Gonzalez, R.E. Woods, “Digital Image Processing Third Edition”, Prentice Hall, pp. 110, 2002.
[4] Sistemas de Visualización, http://webdiis.unizar.es/~SANDRA/MasterIG/SistVisualizacion13-14.pdf.
[5] Antonio Carretero, Metodología didáctica para enseñanza de Geometría Descriptiva basada en un Tutor-Evaluador y un Generador de ejercicios integrados en un entorno de propósito constructivo general, pp. 123, 2001.
[6] Transformaciones Geométricas, http://catarina.udlap.mx/u_dl_a/tales/documentos/mcc/cruz_m_ia/capitulo3.pdf
[7] Fundamentos de la Visión Tridimensional 3D, Universidad Rey Juan Carlos.
[8] Ginés García Mateos, Procesamiento Audiovisual, Tema 4, Transformaciones Geométricas.
[9] Transformaciones Geométricas en OpenGL. http://deckerix.com/blog/transformaciones-geometricas-en-opengl/
[10] OpenGL http://es.wikipedia.org/wiki/OpenGL.
[11] Processing, http://www.processing.org/
[12] Jordi Linares, Gráficos por Computador 3D con Processing. http://users.dsic.upv.es/~jlinares/grafics/processing_spa_7.pdf
[13] Chang Hong Liua, Face recognition with perspective transformation, 2003.
[14] Bing-Fei Wu, Robust Image Measurement and Analysis Based on Perspective Transformations, 2006
[15] Yongtae Do, On the Neural Computation of the Scale Factor in Perspective Transformation Camera Model, 2013.
[16] F. Bonin-Font, A. Burguera, Concurrent visual navigation and localization using inverse perspective transformation, 2012.
[17] Hua Zhang, Changqian Zhu, Qiang Peng, and Jim X. Chen, Using Geometric Algebra for 3D Linear Transformations, 2006.
[18] Eduardo Bayro Corrochano and Eduardo Vazquez Santacruz, A Geometric Radial Basis Function Network for Tracking Variant 3D Transformations, 2007.
[19] G. V. Puskorius and L. A. Feldkamp, Camera calibration method based on a linear perspective transformation error model, 1980.
[20] Hiroaki Nishino, Tsuneo Kagawa, An IEC-Based 3D Geometric Morphing System, 2003.
[21] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao, Estimation of Number of People in Crowded Scenes Using Perspective Transformation, 2001.