

- Búsqueda por Comunicación y Grupo Temático
- Búsqueda por Autor
- Comités (Créditos Publicación)
- Patrocinadores

5 · 6 · 7 de Septiembre · HUELVA
XXVIII Jornadas de Automática

Salir



CEA
comité
español de
automática



Universidad
de Huelva



RECONSTRUCCIÓN TRIDIMENSIONAL DE ESCENAS CON UN PAR ESTEREOSCÓPICO DE CÁMARAS

Blanes F., Jiménez L.M., Puerto R., Ñeco R.P., Reinoso O.
Departamento de Ingeniería. Sistemas industriales
Escuela Politécnica
Universidad M
luis.jimen

Resumen

En el amplio campo de la visión artificial destaca por sí misma la parte de reconstrucción tridimensional por el simple hecho de recrear el proceso de visión “tridimensional” humano. Como bien es sabido, el proceso de reconstrucción tridimensional a partir de un par estereoscópico de cámaras consta de tres fases bien diferenciadas: **calibración, correspondencia y reconstrucción**. El presente artículo recoge aspectos fundamentales sobre el estudio, comparación e implementación de algoritmos de calibración, tanto euclídea como proyectiva, así como de reconstrucción, obviando la fase de correspondencia estereoscópica.

Palabras clave: visión artificial, visión estereoscópica, calibración, geometría proyectiva, reconstrucción, patrón.

1. INTRODUCCIÓN

Generalmente el concepto **reconstrucción 3D** suele llevar a pensar en algo más cercano al campo de la realidad virtual que al de la visión por computador. En realidad la primera es consecuencia de la segunda, pero, ¿qué es la reconstrucción tridimensional a partir de un par estereoscópico?

En la actualidad existen muy diversos métodos para extraer la información espacial de una escena (ultrasonidos, infrarrojos, luz láser estructurada, . . .), no obstante, en el campo de la visión artificial es donde se dispone del método más fácilmente asumible por la mente humana —la **reconstrucción estereoscópica**— debido a su extrema similitud con el proceso de visión humano.

Vemos en 3 dimensiones, aparte de por otros elementos psicológicos —superposición, tamaño familiar, perspectiva, . . .— gracias a que poseemos dos ojos separados una cierta distancia (*disparidad binocular*), lo que posibilita que en cada uno de ellos la imagen percibida sea ligeramente diferente (Figura 1). De esa forma, es en el cerebro donde se integran en una única imagen, utilizando éste, dichas diferencias para obtener la información espacial de la escena.

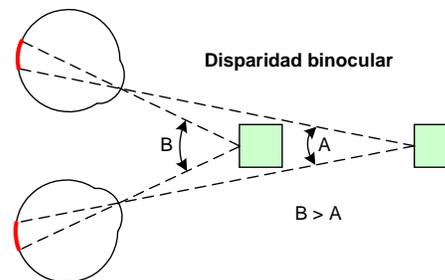


Figura 1: Disparidad binocular.

El proceso de reconstrucción mediante un par estereoscópico de cámaras imita el sistema de visión 3D humano en su parte final pudiendo identificarse las siguientes fases:

1. *Calibración*. Obtención de los parámetros intrínsecos y de distorsión de cada cámara en particular.
2. *Correspondencia*. Identificación de la proyección correspondiente en la imagen contraria.
3. *Reconstrucción*. Cálculo de la coordenada espacial a partir de la disparidad en las proyecciones.

Este artículo se presenta como resumen del Proyecto de fin de carrera (PFC) del mismo título realizado entre 2003 y 2005 en la Universidad Miguel Hernández [1] que comprende íntegramente tanto la primera como la tercera parte del proceso de reconstrucción arriba definido.

El mismo versa sobre el estudio, comparación e implementación de algoritmos de calibración, tanto euclídea como proyectiva, para la reconstrucción de escenas tridimensionales a partir de las imágenes captadas por un par estereoscópico de cámaras no calibradas, constituyendo el resultado más importante el programa *CalUniGraf*, un entorno integrado de calibración, calibración proyectiva y reconstrucción basado en la librería GPL **OpenCV** de Intel[®].

2. MARCO Y OBJETIVOS

El trabajo está integrado en un proyecto de orden superior que engloba los elementos más destacados en cuanto a **fusión multisensorial para navegación autónoma de robots móviles en entornos no estructurados**. En concreto, trata sobre el desarrollo de comportamientos sensoriales (ultrasonidos, infrarrojos, láser, táctiles y visión estéreo) cooperativos para permitir con éxito un desplazamiento autónomo a un robot móvil dentro de un mapa desconocido a priori.

De ese modo, el proyecto realizado se centra en la parte del desarrollo de los algoritmos pertinentes para el control visual a partir del par estereoscópico de cámaras. Por ello, el primer objetivo del mismo es la implementación de dichos procedimientos en el computador interno del robot utilizado (*S.O. Linux RedHat 6.2*).

Sin embargo, debido a diferentes incompatibilidades entre librerías, se pensó en generar un software gráfico GPL —bajo Windows— que integrase todos los aspectos del proceso de reconstrucción tridimensional a partir de un par estereoscópico de cámaras no calibradas. Ello, por ende, se convirtió en el objetivo principal del proyecto, relegando al primero a una fase posterior de simple migración.

Además dicho programa pretende cubrir el nicho existente en cuanto a entornos integrados de código libre¹ estando basado completamente en la librería OpenCV de licencia GPL.

3. EQUIPO Y MATERIAL UTILIZADO

El robot sobre el que se encuentra el par estereoscópico utilizado es el **robot RWI B21r de IRobot®** (Figura 2). Se trata de un robot avanzado y altamente capaz diseñado para entornos de I+D que cubre un amplio rango de aplicaciones robóticas de interior, construido de forma modular y pensado para programación distribuida. Consta de 3 partes principales:

- *Consola*: Se encuentra en la parte superior y soporta además del par estéreo, el receptor Wi-Fi, el sistema de control rFLEX y los botones de emergencia.
- *Cuerpo*: Contiene los PC's del robot, el sistema de potencia, cableado y sensores ultrasonidos, infrarrojos y táctiles.



Figura 2: Robot RWI B21r de IRobot®.

- *Base*: Comprende las ruedas, el sistema mecánico y los componentes de la dirección, las baterías, los motores, sensores táctiles y algunas funciones de distribución de energía.

El robot incorpora el software **Mobility** [5] de IRobot® desarrollado para programación de sistemas distribuidos de control de robots. Éste está basado en un cierto número de servidores diseñados para el control del robot vía red y se encuentra implementado a través de una serie de objetos CORBA que nos ofrecen una interfaz para invocar llamadas remotas.

El par estereoscópico utilizado consta de dos partes fundamentales: las cámaras empleadas y su soporte. Éste último (**Pan-Tilt Unit PTU-46-17.5 Directed Perception®**) simplemente se trata de una plataforma móvil de dos grados de libertad —cabeceo y guiñada— con el fin de posibilitar un mayor campo de visión en situaciones comprometidas en cuanto a navegación autónoma.

Las cámaras utilizadas (**XC-999P Sony®**), por otra parte, fueron cámaras CCD color miniaturizadas ($22 \times 22 \times 120$ mm) de resolución media (752×582) funcionando según el estándar CCIR y el sistema PAL.

Antes de pasar a describir el desarrollo del proyecto realizado, cabe hacer una mínima reseña a la librería sobre la que se basó el mismo: la librería GPL **OpenCV** (*Open Source Computer Vision*

¹**Programas comerciales**: REALISE de la UE y TELE2 de Inria®



Figura 3: Par estereoscópico utilizado.

Library). Se trata de una librería de alto nivel escrita en C/C++ que absorbe la anterior librería de Intel[®] **IPL** (*Image Processing Library*) de bajo nivel. Ésta última incorpora funciones básicas de filtrado, umbralización, . . . mientras que OpenCV implementa algoritmos más avanzados sobre calibración, optical flow, análisis de formas, . . .

4. DESARROLLO DEL PROYECTO

Previamente al inicio de la etapa de desarrollo de la aplicación principal del proyecto fue necesaria una fase previa en la cual hubo que realizar la configuración tanto del PC del robot y sus servidores (Linux RedHat 6.2) así como un acondicionamiento de los PC's externos utilizados como estaciones de desarrollo. Todo ello coordinado a través de una red inalámbrica.

Además, tras observar las características² del servidor de vídeo proporcionado por IRobot, se decidió reprogramarlo con el fin de hacerlo más flexible y adaptarlo convenientemente a las necesidades del proyecto. Así mismo, se desarrolló un programa básico de captura para la obtención de las imágenes experimentales en el que se incluyó

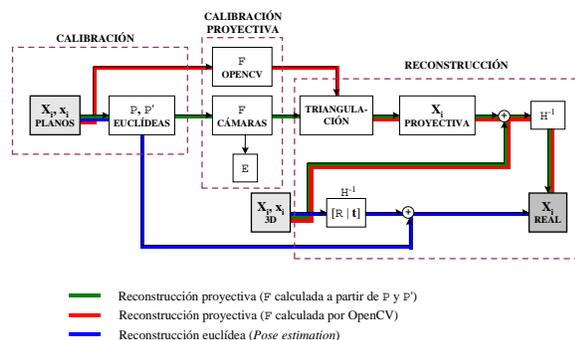


Figura 4: Proceso de reconstrucción utilizado.

²Brillo, saturación, . . . fijas. Resolución fija 320×240 → insuficiente.

Fijándonos en la Figura 4, y tras obviar la parte de correspondencia, el proceso de reconstrucción 3D queda compuesto por 3 fases: *calibración*, *calibración proyectiva* y *reconstrucción 3D*.

4.1. PROCESO DE CALIBRACIÓN

El objetivo de la fase de calibración de cada una de las 2 cámaras CCD por separado era proporcionar, a partir de un cierto número de imágenes de un patrón de calibración 2D situado a diferentes distancias, los parámetros intrínsecos de cada una de ellas así como sus respectivos parámetros de distorsión.

En primer lugar, hubo que diseñar el tipo de **patrón de calibración** a utilizar. Basándonos en la función de OpenCV `cvFindChessBoardCornerGuesses` —detecta las esquinas de las “casillas de un tablero de ajedrez” con resolución subpíxel— se decidió

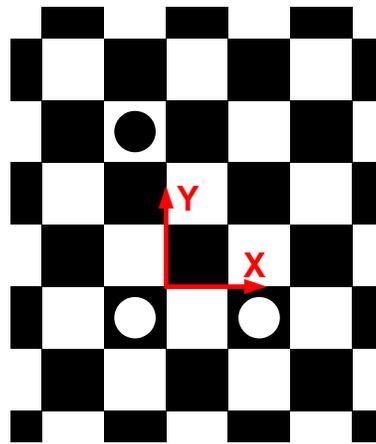


Figura 5: Patrón de calibración 2D.

Para poder identificar un origen en el patrón de calibración sobre las imágenes tomadas del mismo, se situaron los dos círculos blancos y el círculo negro. De tal forma, que siempre, debido a que una transformación proyectiva conserva la razón doble de distancias, el círculo negro definiría el eje Y y el círculo blanco el eje X.

Una vez diseñado el patrón de calibración, para poder hacer uso de la función de OpenCV `cvCalibrateCamera`, se procede a la obtención de correspondencias 3D-2D. Para ello, en primer lugar, es necesario obtener la posición de los centros de los círculos del patrón para definir unívocamente el sistema de coordenadas del mundo sobre el patrón: 1.- Filtro de bordes de Canny 2.- Búsqueda de contornos (`cvFindContours`) 3.- Algoritmo de selección de círculos (compacidad, área, similitud con elipse, . . .)

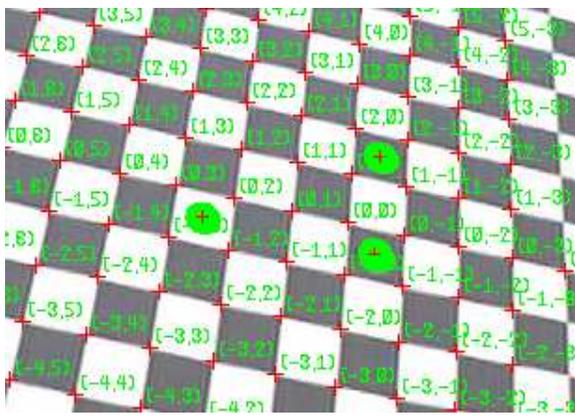


Figura 6: Resultado de la identificación de correspondencias.

A continuación, se procede a la estimación de las esquinas del patrón de calibración sobre la imagen haciendo uso de la función antes comentada `cvFindChessBoardCornerGuesses` restando únicamente la búsqueda de la correspondencia con los puntos espaciales.

Para ello, se realiza un proceso de *matching* de las esquinas estimadas sobre las posiciones de una “rejilla” virtual creada a partir del origen del patrón de calibración atendiendo a las direcciones estimadas de los ejes X e Y. De esa forma se alcanza un resultado como el de la Figura 6.

Finalmente, el proceso de calibración termina al lanzar la función `cvCalibrateCamera`—algoritmo DLT normalizado más optimización no lineal (*Levenberg-Marquardt*)—obteniéndose la **matriz de calibración** K y los **parámetros de distorsión** de la cámara. En concreto,

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

y el modelo de distorsión utilizado (p_1, p_2 distorsión tangencial; k_1, k_2 distorsión radial), con (u, v) las coordenadas originales de un punto sin distorsión y (\tilde{u}, \tilde{v}) las coordenadas reales del punto,

$$\begin{aligned} \tilde{u} &= u + (u - c_x) \left[k_1 r^2 + k_2 r^4 + 2p_1 y + p_2 \left(\frac{r^2}{x} + 2x \right) \right] \\ \tilde{v} &= v + (v - c_y) \left[k_1 r^2 + k_2 r^4 + 2p_2 x + p_1 \left(\frac{r^2}{y} + 2y \right) \right] \end{aligned}$$

4.2. CALIBRACIÓN PROYECTIVA

El objetivo principal del proceso de calibración proyectiva era la estimación de la **matriz fundamental** F , matriz que recoge la posición y orientación de una cámara con respecto a la otra

así como los parámetros intrínsecos de ambas. Además, como objetivos secundarios derivados se pretendió calcular la posición de los **epipolos** a la vez que una representación de las **líneas epipolares** sobre las imágenes.

Para ello, se implementaron dos métodos diferentes:

1. a partir de correspondencias \mathbf{x}_i y \mathbf{x}'_i
2. a partir de las matrices de cámara P y P' .

Cabe decir que ambos métodos parten de la identificación de las correspondencias 3D-2D según el método utilizado en la parte de calibración.

El **primer método** se inicia con la identificación de los puntos coincidentes en ambas imágenes de entre todos aquellos correctamente estimados según el proceso anteriormente citado. Una vez hecho esto se procede al cálculo de la matriz fundamental F usando la función de OpenCV `cvFindFundamentalMat`. Hay que comentar que el algoritmo que utiliza dicha función es el *algoritmo de 8 puntos normalizado* basado en la relación

$$\mathbf{x}'_i{}^\top F \mathbf{x}_i = 0 \quad (1)$$

Por otro lado, el **segundo método** produce una derivación algebraica de la matriz F a partir de las matrices de cámara calculadas a partir de las matrices de calibración K y K' y los parámetros extrínsecos obtenidos de las correspondencias 3D-2D de la presente fase. De ese modo, situando el origen del sistema de coordenadas del mundo en el centro óptico de la primera cámara,

$$P = K [I | \mathbf{0}] \quad P' = K' [R | \mathbf{t}] \quad (2)$$

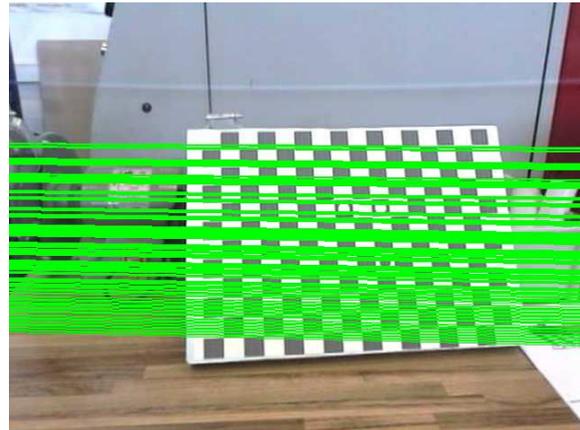


Figura 7: Líneas epipolares “horizontales” en una configuración de ejes paralelos.

y a partir de ellas, calculando los epipolos como la imagen del centro óptico de la otra cámara,

$$\mathbf{e} = \mathbf{P} \begin{pmatrix} -\mathbf{R}^T \mathbf{t} \\ 1 \end{pmatrix} = \mathbf{K}\mathbf{R}^T \mathbf{t} \quad \mathbf{e}' = \mathbf{P}' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \mathbf{K}'\mathbf{t} \quad (3)$$

entonces, es posible calcular la matriz fundamental \mathbf{F} según varias expresiones, entre las cuales se eligió la siguiente:

$$\mathbf{F} = [\mathbf{e}']_{\times} \mathbf{K}'\mathbf{R}\mathbf{K}^{-1} \quad (4)$$

Una vez obtenida la matriz fundamental, es sencillo obtener los epipolos

$$\mathbf{F}^T \mathbf{e}' = \mathbf{0} \quad \mathbf{F}\mathbf{e} = \mathbf{0} \quad (5)$$

así como las líneas epipolares de ciertos puntos de las imágenes. En nuestro caso, debido a que las cámaras estaban situadas en configuración de ejes paralelos, teóricamente los epipolos estaban en el infinito y las líneas epipolares debían ser horizontales, tal como puede observarse en la Figura 7.

4.3. RECONSTRUCCIÓN

Finalmente, el proceso de reconstrucción tiene como objetivo la estimación de las coordenadas espaciales de un punto a partir de sus proyecciones correspondientes. Objetivo que coincide con el inicial del proyecto. Además se pretende la obtención de las matrices de cámara reales \mathbf{P}_{real} y $\mathbf{P}'_{\text{real}}$.

En primer lugar, con el fin de obtener información espacial, se requirió la construcción de un patrón de calibración 3D. Con el fin de aprovechar los

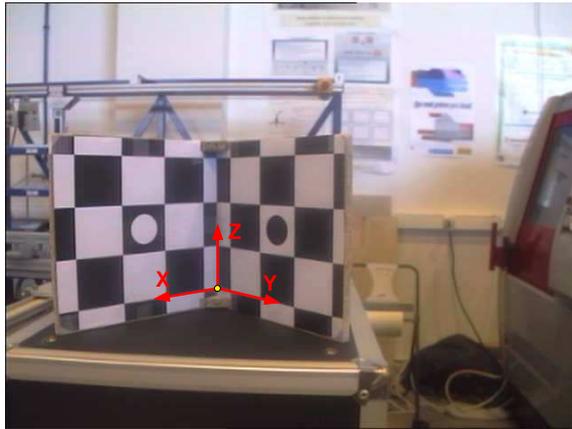


Figura 8: Sistema de coordenadas asociado al patrón de calibración 3D diseñado.

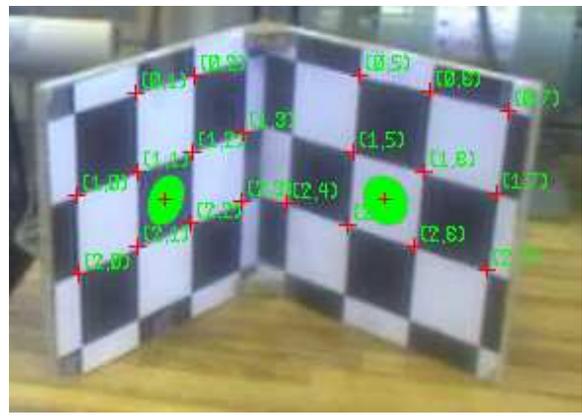


Figura 9: Resultado de la identificación de correspondencias del patrón de calibración 3D.

primer paso es la obtención de un cierto número de correspondencias 3D-2D para su uso posterior. Para ello, el algoritmo utilizado es completamente análogo al desarrollado para las fases anteriores, con las mínimas variaciones necesarias: 1.- Identificación de los círculos en el patrón 2.- Estimación de las esquinas del patrón (`cvFindChessBoard.`) 3.- Algoritmo de correspondencia con los puntos espaciales.

De esa forma, el resultado que se obtiene, similar al de la Figura 6 del proceso de calibración, puede verse en la Figura 9.

Llegados a este punto, tras la obtención de la matriz fundamental en el proceso de calibración proyectiva, se tiene una *reconstrucción proyectiva*, la cual para migrarla a una **reconstrucción métrica** requiere del cálculo de una *matriz de transformación proyectiva* \mathbf{H} tal que

$$\mathbf{P}_{\text{real}} = \mathbf{P}\mathbf{H}^{-1} \quad \mathbf{P}'_{\text{real}} = \mathbf{P}'\mathbf{H}^{-1} \quad (6)$$

De esa forma, para el cálculo de dicha matriz \mathbf{H} se plantearon dos métodos, tal como puede verse en la Figura 4:

1. **Reconstrucción proyectiva.** Partiendo de la matriz fundamental \mathbf{F} .
2. **Pose estimation.** Reconstrucción euclídea directa a partir de \mathbf{x}_i y \mathbf{x}'_i .

El primero de ellos se inicia con la búsqueda de los puntos coincidentes entre los identificados en ambas imágenes para después obtener por *triangulación* el punto espacial correspondiente \mathbf{X}_i .

El método de triangulación utilizado fue el **método de triangulación óptimo** [3] (`cvOptimalTriangF`). En ese momento, se procedió al cálculo de la matriz \mathbf{H} de forma algebraica por mínimos cuadrados (SVD) partiendo

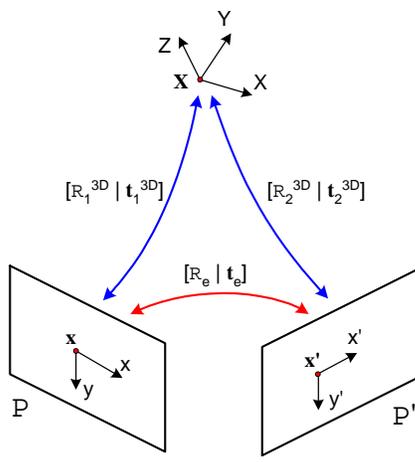


Figura 10: Pose estimation.

de la relación entre el punto real conocido \mathbf{X}_{Mi} y el estimado \mathbf{X}_i ,

$$\mathbf{X}_{Mi} = \mathbf{H}\mathbf{X}_i \quad (7)$$

El segundo método utilizado (*Pose estimation*) en todo momento prescinde de una reconstrucción proyectiva intermedia, pasando de forma directa a una reconstrucción métrica a través del cálculo de los parámetros extrínsecos en cada cámara.

Atendiendo a la Figura 10 y según el desarrollo de [1] se llega a la conclusión que

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{R}_1^T & \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (8)$$

5. RESULTADOS: CalUniGraf

Observando los objetivos planteados, cabe señalar a la aplicación desarrollada **CalUniGraf** como el principal resultado conseguido. Se trata de un programa sencillo, intuitivo, flexible y de trabajo si-

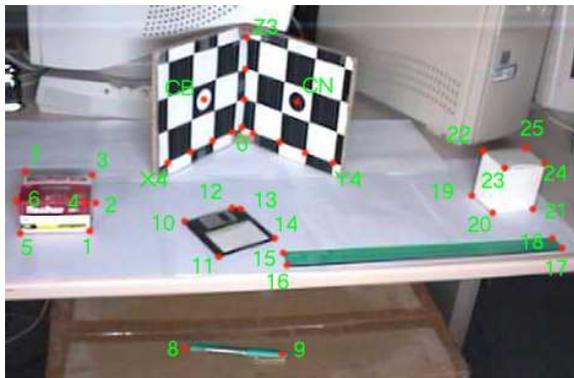


Figura 11: Escena reconstruida en Figura 12 y Figura 13.

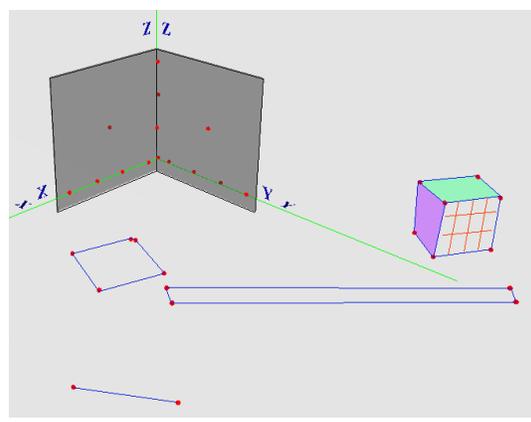


Figura 12: Escena reconstruida (Isométrica).

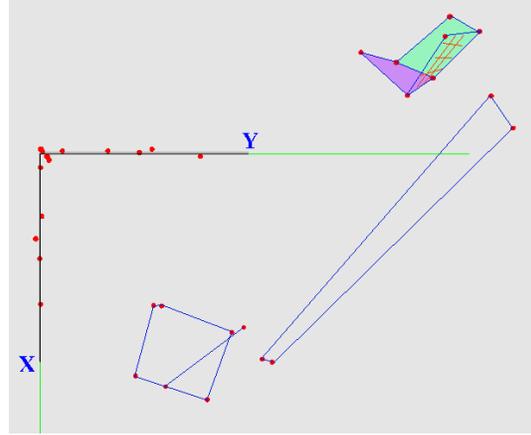


Figura 13: Escena reconstruida (Planta).

multáneo, con ayudas inmediatas sobre cada comando,... Incorpora todos los elementos necesarios en cuanto a calibración de cámaras monocromas y en color. La parte de calibración proyectiva ha sido desarrollada con éxito vía dos métodos diferentes, de la misma forma que la fase de reconstrucción adjunta *reconstrucción proyectiva* y *pose estimation*.

Además el programa incorpora una herramienta visual dinámica de representación de las escenas reconstruidas programada en lenguaje VRML 2.0. De esa forma, se permite navegar tridimensionalmente sobre los puntos espaciales reconstruidos.

Antes de cerrar el capítulo de resultados es de rigor comentar el hecho de que se realizaron un gran número de experimentos de los cuales se da un intensivo análisis en [1].

6. CONCLUSIONES

Para finalizar, hay que destacar que se realizó correctamente el diseño de la aplicación objetivo *CalUniGraf*: un programa gráfico integrado de

calibración, calibración proyectiva y reconstrucción tridimensional de escenas. Además es imprescindible comentar el hecho de su cómoda portabilidad a un sistema Linux al haber sido desarrollado utilizando librerías de código libre (OpenCV) y fácilmente portables (Qt).

Este proyecto de fin carrera hay que tomarlo seguramente como la puerta u origen de otros nuevos trabajos, tanto basados en él como posibles mejoras del mismo: realidad aumentada, autocalibración,...

Referencias

- [1] Blanes, F., (2005) *Reconstrucción 3D de escenas con un par estereoscópico de cámaras*, PFC Universidad Miguel Hernández.
<http://isa.umh.es/pfc/rmvision>
- [2] Faugeras, O., (1993) *Three-Dimensional Computer Vision*, The Massachusetts Institute of Technology (MIT) Press.
- [3] Hartley, R., Zisserman, A., (2000) *Multiple View Geometry in computer vision*, Cambridge University Press.
- [4] Intel Corporation, (2001) *Open Source Computer Vision Library Reference Manual*, Intel Corporation.
- [5] iRobot Corporation, (2002) *Mobility Robot Integration Software User's Guide*, iRobot Corporation.
- [6] Xu, G., Zhang Z., (1996) *Epipolar Geometry in Stereo, Motion and Object Recognition*, Kluwer Academic Publishers.