

Homograf, un programa para la resolución de homografías 2D En Autocad V.:2002.

F. Irlés ¹, L. M. Jiménez ², M. A. Oliva ³, E. Velasco ⁴.

Universidad Miguel Hernández de Elche. Av. Ferrocarril, s/n. Elche-Alicante-España.

¹ Paco.Irles@umh.es

² luis.jimenez@umh.es

³ ma.oliva@umh.es

⁴ emilio.velasco@umh.es

Palabras Clave: Sistemas de Representación – CAD, Instrumentación, Métodos numéricos, Ensayos y verificaciones.

Resumen. Este artículo presenta un algoritmo para la resolución de homografías 2D así como su aplicación como una herramienta de Autocad. Este trabajo está basado en la geometría proyectiva de múltiples vistas tomadas con una cámara. La matriz de colineación entre dos planos proyectivos se calcula a partir de cuatro pares de puntos homólogos. Entre las aplicaciones están: fotogrametría plana con una sola fotografía oblicua, generación de anaglifos, fotocomposición, creación y distorsión de perspectivas, corrección cartografía, etc. En el campo industrial estamos investigando en sistemas de guiado en robots, identificación y control de piezas, filmación a alta velocidad con cuadros oblicuos.

Abstract. This paper presents an algorithm that solves 2D homographies and its application as an Autocad tool. This work is based on projective geometry of multiple view camera snapshots. A collineation matrix between two projective planes is calculated from four pairs of matching points. The implemented applications include: bidimensional photogrametry with a single oblique picture, anaglifos generation, photocomposition, generation and distortion of perspectives, cartography correction, etc. In the industrial field is under investigation systems for robot guiding, parts identification, and high speed filming with oblique point of views.

Antecedentes y motivación.

Hoy en día son múltiples los programas de fotogrametría comercializados en el mercado, los más completos que utilizan dos o más fotogramas para montar pares estereoscópicos son extremadamente costosos y de cierta dificultad de manejo por la cantidad de datos que se requieren. La fotogrametría estática y dinámica (con dos o más secuencias de video) queda por tanto reservada a unos pocos técnicos que trabajen en empresas que puedan soportar de forma rentable esta inversión. Nuestro trabajo va orientado a trabajos de fotogrametría con una sola imagen o una sola secuencia de video, donde solo se busquen medidas de un plano real y resulte innecesario montar pares estereoscópicos. En el ejercicio profesional de los arquitectos es muy común requerir de levantamientos de planos de fachadas con escasa precisión, por cuestiones de diseño. Una gran ayuda se consigue mediante fotografías, normalmente proyecciones cónicas (aproximadamente). En aplicaciones industriales resulta interesante para su utilización para desarrollo de sistemas de guiado en robots que trabajan en espacios predominantemente 2D, identificación y control de piezas planas a través de invariantes proyectivos, así como en la instrumentación mediante una sola cámara de video en ensayos de carácter 2D, y cuadros oblicuos variables.

El autor ⁽¹⁾ había desarrollado trabajos en esta línea en colaboración con el arquitecto Ramón Maestre, abordando el problema desde el punto de vista de la geometría plana clásica y programándolo en Autolisp, bien desde su misma base, la hología, o como una macro que empleaba la orden tablero de Autocad con una calibración por cuatro puntos. Estas soluciones adolecen de

lentitud y de un farragoso tratamiento para multitud de casos particulares. Es por ello que dentro de los trabajos de doctorado en "Tecnologías Industriales" en la U.M.H. se opta por abordar el problema desde la geometría proyectiva, logrando un algoritmo basado en calculo matricial, robusto para toda casuística e independiente de comandos de Autocad, programando en C para lograr una mayor rapidez de la aplicación.

Desarrollo teórico.

En un primer lugar se analizó la problemática de poder medir sobre fotografías distancias en un plano conocido y se resolvió a través de dos homologías planas, invirtiendo el proceso de obtención de cónicas descrito en los tratados de geometría descriptiva a través de homologías planas de las vistas diédricas.

Este sistema adolecía de los problemas típicos de la geometría convencional, fallaba en casos de paralelismos debido a las intersecciones de las rectas en puntos impropios y en consecuencia resultaba extremadamente complejo de programar y depurar.

Es por ello que los autore optan por buscar otro algoritmo que resuelva el problema, esta solución se apoyo en la posibilidad que presenta AutoCAD de calibrar los tableros digitalizadores mediante: 2 pares de puntos (euclídea), 3 pares de puntos (afín) y 4 ó más pares de puntos (proyectiva). La transformación proyectiva por cuatro puntos consigue resultados idénticos a otro sistema.

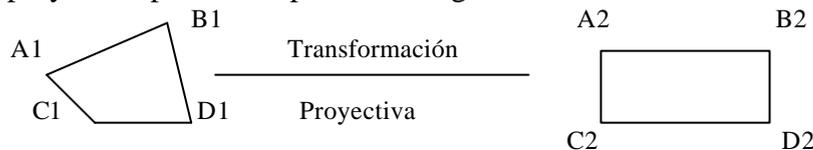


Figura 1. Tranformación proyectiva

Con esto se desarrolla un programa (primero el AutoLISP y luego es portado a C), donde se dan diversas opciones de entrada para definir la transformación proyectiva, convirtiéndose todas a cuatro pares de puntos, que con el comando de autocad "tablero" se obtiene la matriz 3x3 de transformación.

La aplicación de esta transformación requería multiplicar las coordenadas homogéneas de los puntos dato por la matriz y obtenía unas coordenadas proyectivas resultado que se convertían en euclídeas dividiendo por la coordenada Z.

Este programa se comercializo junto con un libro en el año 2000, sin embargo pronto aparecería una nueva necesidad entre los usuarios: el programa no funciona en AutoCAD 2000 debido a que éste no permite la ejecución del comando tablero si no existe físicamente, lo que ocurre en la mayoría de los casos al trabajarse con fotografías escaneadas. Se requiere así determinar la matriz de la transformación a partir de los datos directamente.

La transformación proyectiva que nos ocupa se intento resolver mediante operaciones geométricas de *warping*. Para resolver la transformación se han seguido dos pasos, ya que el primero fue infructuoso.

En primer lugar nos basamos en una aplicación existente de *warping*, la cual aparentemente resolvía el problema. Analizado el algoritmo del código fuente en MatLab, y programado en Autolisp, resulta ser una transformación mediante una matriz 4x4 basada en 4 pares de puntos

Los resultados obtenidos en un primer momento se parecen a la solución correcta (totalmente en el contorno del cuadrilátero dato y menos hacia el interior [figura 2])., resultando valido para

representaciones pictóricas pero no para métricas como se pretendía. Es por ello que se descarta este sistema.

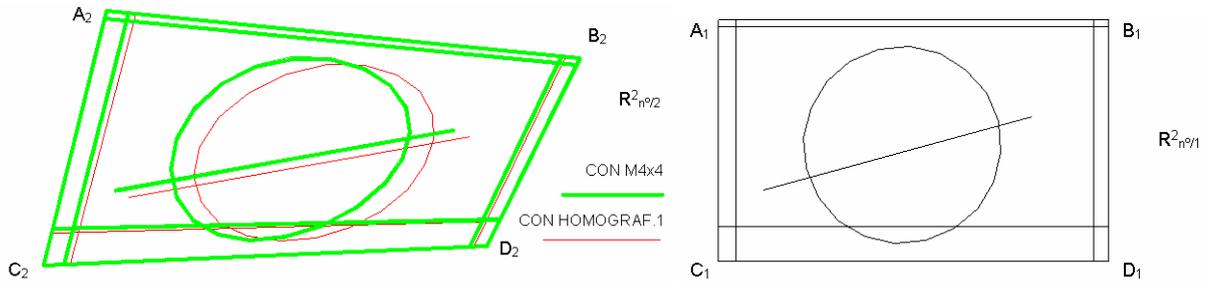


Figura 2 Transformación de warping.

La solución definitiva la encontramos empleando Geometría Proyectiva, en concreto la ecuación que se emplea para correlar pares de puntos en pares estereoscópicos. Partiendo de la ecuación matricial con la que se define la colineación, y partiendo de unos datos para definir la transformación proyectiva que consisten en cuatro pares de puntos homográficos, se obtiene la ecuación matricial de colineación: $A \tilde{e}_i = k_i \tilde{x}_i$. Donde:

- $A \tilde{}$ es la matriz de colineación;
 - \tilde{e}_i son los puntos de partida en coordenadas homogéneas;
 - \tilde{x}_i son los puntos de destino en coordenadas proyectivas y
 - λ_i son coeficientes de escalado con i de 1 a 4.
- (Nota: La terminación en $\tilde{}$ indica en coordenadas proyectivas).

Datos:	<u>Espacio de $R^2_{n^o/1}$</u>	<u>Espacio de $R^2_{n^o/2}$</u>
	A1=(x_{A1}, y_{A1}) C1=(x_{C1}, y_{C1})	A2=(x_{A2}, y_{A2}) C2=(x_{C2}, y_{C2})
	B1=(x_{B1}, y_{B1}) D1=(x_{D1}, y_{D1})	B2=(x_{B2}, y_{B2}) D2=(x_{D2}, y_{D2})

Tomando valores en la matriz a partir de los datos planteamos las siguientes ecuaciones:

$$\left. \begin{aligned} A \tilde{\cdot} A1 \tilde{=} \lambda_1 \cdot A2 \tilde{=} \\ A \tilde{\cdot} B1 \tilde{=} \lambda_1 \cdot B2 \tilde{=} \\ A \tilde{\cdot} C1 \tilde{=} \lambda_1 \cdot C2 \tilde{=} \\ A \tilde{\cdot} D1 \tilde{=} \lambda_1 \cdot D2 \tilde{=} \end{aligned} \right\} ? \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \\ \dots & & \dots \end{pmatrix} \cdot \begin{pmatrix} x_{A1} \\ y_{A1} \\ 1 \\ \dots \end{pmatrix} = \begin{pmatrix} \lambda_1 x_{A2} \\ \lambda_1 y_{A2} \\ \lambda_1 \\ \dots \end{pmatrix} ?$$

? $\left. \begin{aligned} x_{A1} a_{11} + y_{A1} a_{12} + 1 a_{13} &= \lambda_1 x_{A2} \\ x_{A1} a_{21} + y_{A1} a_{22} + 1 a_{23} &= \lambda_1 y_{A2} \\ x_{A1} a_{31} + y_{A1} a_{32} + 1 &= \lambda_1 \\ \dots & \dots \dots \end{aligned} \right\}$ Donde estamos asumiendo que los escalares λ_i absorben el factor de escado a_{33} de toda colineación. Operando con los cuatro pares de puntos, podemos obtener 12 ecuaciones, tres por cada par, que ordenadas convenientemente como un producto matricial quedaría:

$$\begin{pmatrix} x_{A1} & y_{A1} & 1 & 0 & 0 & 0 & 0 & 0 \\ x_{B1} & y_{B1} & 1 & 0 & 0 & 0 & 0 & 0 \\ x_{C1} & y_{C1} & 1 & 0 & 0 & 0 & 0 & 0 \\ x_{D1} & y_{D1} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{A1} & y_{A1} & 1 & 0 & 0 \\ 0 & 0 & 0 & x_{B1} & y_{B1} & 1 & 0 & 0 \\ 0 & 0 & 0 & x_{C1} & y_{C1} & 1 & 0 & 0 \\ 0 & 0 & 0 & x_{D1} & y_{D1} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{A1} & y_{A1} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{B1} & y_{B1} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{C1} & y_{C1} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{D1} & y_{D1} \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix} = \begin{pmatrix} \lambda_1 x_{A2} \\ \lambda_1 y_{A2} \\ \lambda_1 \\ \lambda_2 x_{B2} \\ \lambda_2 y_{B2} \\ \lambda_2 \\ \lambda_3 x_{C2} \\ \lambda_3 y_{C2} \\ \lambda_3 \\ \lambda_4 x_{D2} \\ \lambda_4 y_{D2} \\ \lambda_4 \\ \lambda_1 - 1 \\ \lambda_2 - 1 \\ \lambda_3 - 1 \\ \lambda_4 - 1 \end{pmatrix}$$

De estas 12 ecuaciones con 8 parámetros de la matriz de colineación y 4 factores de escalado como incógnitas, podemos despejar λ_i de las últimas 4 ecuaciones y sustituir sus valores en función de a_{31} y a_{32} en las restantes 8 ecuaciones, con lo que tendremos el problema planteado al ser todas ellas linealmente independientes. Tras esto y ordenándolas quedaría:

$$\begin{pmatrix} x_{A1} & y_{A1} & 1 & 0 & 0 & 0 & -(x_{A1}x_{A2}) & -(y_{A1}x_{A2}) \\ x_{B1} & y_{B1} & 1 & 0 & 0 & 0 & -(x_{B1}x_{B2}) & -(y_{B1}x_{B2}) \\ x_{C1} & y_{C1} & 1 & 0 & 0 & 0 & -(x_{C1}x_{C2}) & -(y_{C1}x_{C2}) \\ x_{D1} & y_{D1} & 1 & 0 & 0 & 0 & -(x_{D1}x_{D2}) & -(y_{D1}x_{D2}) \\ 0 & 0 & 0 & x_{A1} & y_{A1} & 1 & -(x_{A1}y_{A2}) & -(y_{A1}y_{A2}) \\ 0 & 0 & 0 & x_{B1} & y_{B1} & 1 & -(x_{B1}y_{B2}) & -(y_{B1}y_{B2}) \\ 0 & 0 & 0 & x_{C1} & y_{C1} & 1 & -(x_{C1}y_{C2}) & -(y_{C1}y_{C2}) \\ 0 & 0 & 0 & x_{D1} & y_{D1} & 1 & -(x_{D1}y_{D2}) & -(y_{D1}y_{D2}) \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix} = \begin{pmatrix} x_{A2} \\ x_{B2} \\ x_{C2} \\ x_{D2} \\ y_{A2} \\ y_{B2} \\ y_{C2} \\ y_{D2} \end{pmatrix}$$

La resolución de un sistema de ecuaciones se puede abordar de diversas formas, habiéndose elegido la diagonalización de Gauss. Para poder diagonalizar se plantea un problema: tenemos submatrices nulas que impiden aplicar un algoritmo escalonado secuencial, para evitarlo se suman las ecuaciones que se obtendrían con las filas 4 y 5 a 4, 5 y 6 a 5, 6 y 7 a 6, 7 y 8 a 7. Una vez diagonalizada se resuelve el sistema obteniendo los valores a_{ij} de la matriz de colineación.

Conocida la matriz de colineación resta plantear los pasos y las ecuaciones para transformar un punto cualquiera de un espacio euclídeo $R^2_{n^{\circ}/1}$ a el otro de espacio $R^2_{n^{\circ}/2}$, que serán:

- Pasar un punto P1 de coordenadas euclídeas (x_{P1}, y_{P1}) a un punto $P1\tilde{}$ en coordenadas homogéneas $(x_{P1}, y_{P1}, 1)$.
- Multiplicarlo por la matriz de colineación obteniendo el punto $P2\tilde{}$ en coordenadas proyectivas: $A\tilde{\cdot}P1\tilde{}=P2\tilde{}$.
- Pasar el punto $P2\tilde{}$ de coordenadas proyectivas $(x_{p2\tilde{}}, y_{p2\tilde{}}, \lambda)$ a coordenadas euclídeas $(x_{p2\tilde{}}/\lambda, y_{p2\tilde{}}/\lambda)$.

Ejemplos y aplicaciones.

Se pasa a ilustrar con algunas de las aplicaciones desarrolladas con el programa implementado siguiendo el algoritmo descrito

Fotogrametría de una sola fotografía oblicua al plano de fachada. Se inserta la fotografía de la cual se conocen dos medidas, una horizontal y la otra vertical. Se calcan las líneas que interesan y se define la transformación a partir del rectángulo definido por las dos medidas, aplicándola a todas las líneas (figura 3).



Figura 3. Ejemplo de fotogrametría

Montaje fotográfico de un plano de fachada según proyecto. Se toma la fotografía de la calle donde se va a construir, se elimina la zona de la fachada a derruir y se superpone en AutoCAD la imagen vectorial resultado de transformar el plano de fachada al cuadrilátero disponible en el solar (figura 4).

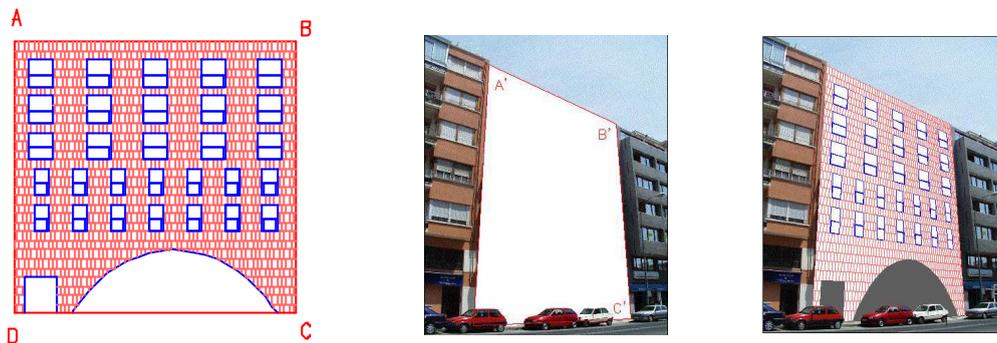


Figura 4. Ejemplo de montaje fotográfico

Obtención de dibujos 2D en perspectiva cónica a partir de las vistas diédricas. Se requiere aplicar la transformación a tantos planos como existan en el edificio (figura 5).

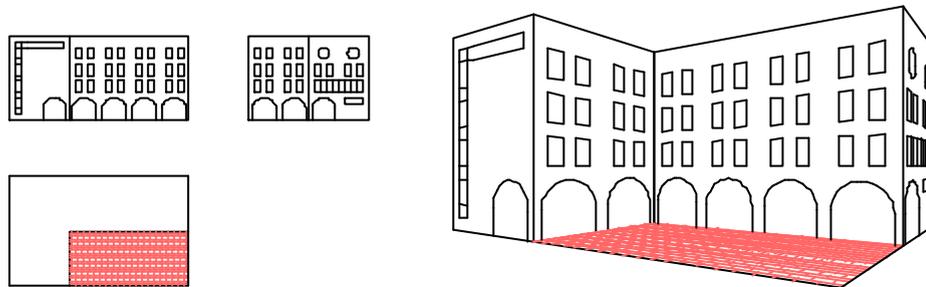


Figura 5. Ejemplo de generación de perspectiva cónica

Fotogrametría de una sola fotografía con corrección de fuga. El procedimiento es idéntico al planteado en el primer ejemplo, pero en este caso se podría resolver el problema sin usar el programa, ya que solo se requiere aplicar una homología para corregir el único punto de fuga. No obstante el uso del programa es ventajoso pues automatiza la ejecución de la homología y simplifica su definición (figura 6).

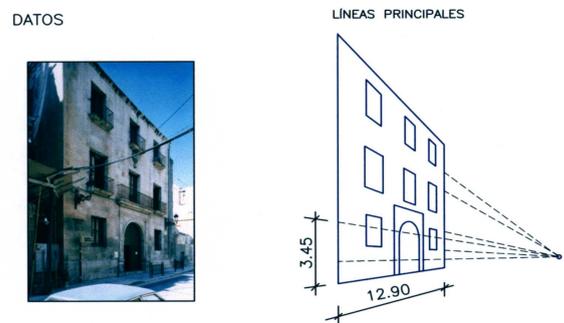


Figura 6 . Ejemplo de corrección de fuga

Obtención de perspectivas cilíndricas ortogonales u oblicuas. El programa posibilita a partir de una proyección ortogonal ó de una cónica la obtención de perspectivas axonométricas isométricas dimétricas ó trimétricas, así como caballeras o militares (estas últimas, por ser oblicuas, no se pueden obtener en AutoCAD 2000) (figura 7).

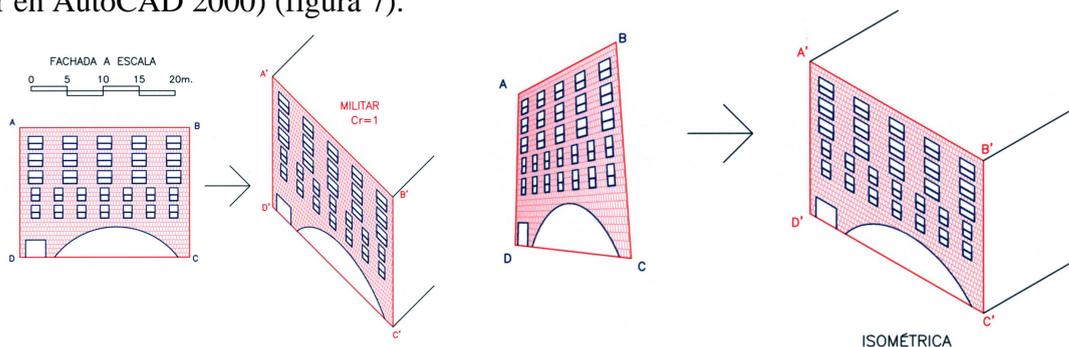


Figura 7 . Obtención de perspectivas cilíndricas y cónicas

Deformación de perspectivas con fines comerciales. En muchas ocasiones los arquitectos deforman las perspectivas con fin de realzar determinados aspectos (figura 8).

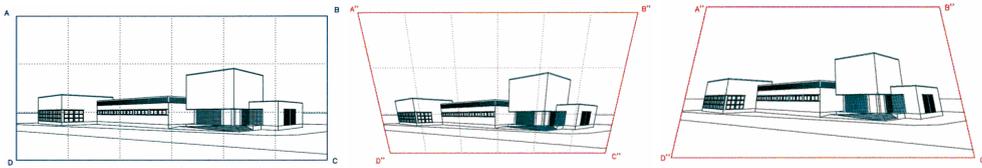


Figura 8. Ejemplo de deformación de perspectiva

Obtención de anaglifos. La obtención de es-te tipo de vista estereoscópica resulta extremadamente rápida y sencilla (figura 9).

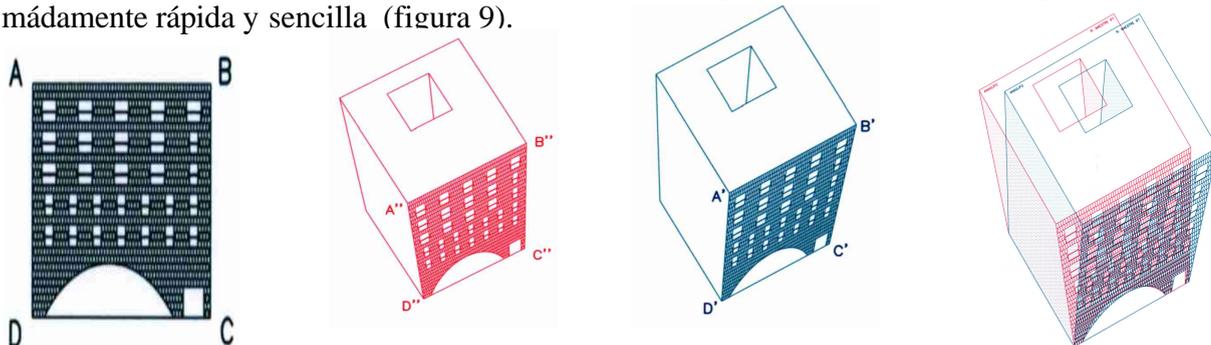


Figura 9. Ejemplo de obtención de anaglifos

Ortogonalización de secuencias de video convencional y de cámara de alta velocidad, empleando el mismo algoritmo pero en un entorno de edición de video en lugar de en Autocad (figura 10).

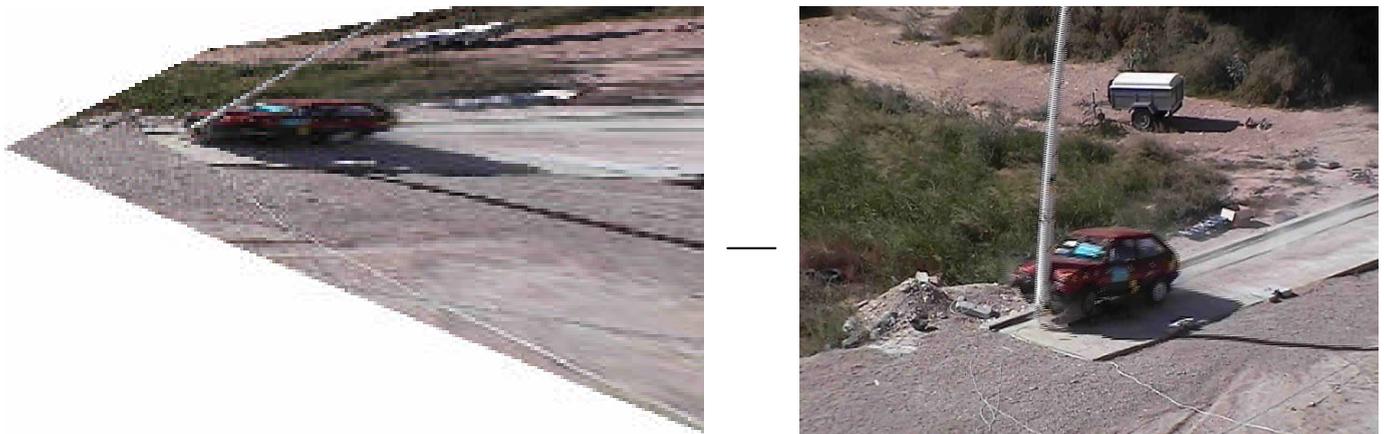


Figura 10. Ejemplo de orthogonalización de secuencias de video

Referencias

- [1] O. Reinoso, J.M. Sebastián. Cursos doctorado de Tecnologías Industriales: Procesamientos Avanzados en Visión Artificial y Visión Tridimensional. www.lorca.umh.es .Ed.: UMH. 2000-01
- [2] R. Maestre, F.Irles. Levantamiento de planos de fachadas a partir de una fotografía y trazado de perspectivas. Ed.: Universidad de Alicante. 2.000.
- [3] Three-Dimensional Computer Vision (Artificial Intelligence). Olivier Faugeras, 1993, Mit Press.
- [4] Multiple View Geometry in Computer Vision. Richard Hartley, Andrew Zisserman, Cambridge University Press 2000.
- [5] F. Irles, C. Senabre, M.Sánchez, E. Velasco. Análisis Comparativo de Mediciones en Ensayos de Impacto sobre Báculo de Alumbrado. IBEM6 - Coimbra 2003. (Aceptado abstract).