

07-005

INCLUDING AERIAL PERSPECTIVE AN AGENT-BASED PARADIGM FOR THE RECONSTRUCTION OF CONICAL PERSPECTIVES

Conesa Pastor, Julián⁽¹⁾; Mula Cruz, Francisco José⁽¹⁾

⁽¹⁾Universidad Politécnica de Cartagena

In a previous work we presented an agent-based architecture which allows obtaining models from graphs with normalon and quasi-normalon typology which represent the conical projection of a model in parallel perspective (with a vanishing point) or oblique perspective (with two vanishing points). In this paper we define a new architecture which broadens our previous work significantly as it also allows dealing with graphs which represent an aerial perspective or with three vanishing points, thereby closing the range of possibilities of the conical perspective. However, the requirements of the analytical calculation needed to deal with this new type of perspective have revealed bottlenecks in the execution of the proposed architecture so far, which means an increase in the computing cost. For this reason, in this paper we present a restructuring of the multi-agent architecture which optimizes that cost. The proposed system distinguishes, in an automatic way, the typology of the perspective of the graph (parallel, oblique or aerial) handling, therefore, any kind of conical perspective in contrast to methodologies proposed by other authors which have always been focused on aerial perspectives.

Keywords: *agent-based system; image reconstruction; conical perspective; image-based modelling*

INCLUYENDO LA PERSPECTIVA AÉREA EN UN SISTEMA BASADO EN AGENTES PARA LA RECONSTRUCCIÓN DE PERSPECTIVAS CÓNICAS

En un trabajo previo presentamos una arquitectura basada en agentes que permite la obtención de modelos a partir de grafos de tipología normalón y quasi-normalón que representan la proyección cónica de un modelo en perspectiva paralela o perspectiva oblicua (de dos puntos de fuga). En este trabajo se define una nueva arquitectura que amplía considerablemente nuestro trabajo previo al permitir tratar además grafos que representan en perspectiva aérea o de tres puntos de fuga un grafo, cerrando así el conjunto de posibilidades de la perspectiva cónica. Sin embargo, las exigencias del cálculo analítico necesario para tratar este nuevo tipo de perspectiva, ha puesto de manifiesto cuellos de botella en la ejecución de la arquitectura hasta el momento propuesta que se traduce en un aumento del coste computacional. Por este motivo en este trabajo se presenta una reestructuración de la arquitectura multi-agente que optimiza dicho coste. El sistema propuesto diferencia de forma automática la tipología de la perspectiva del grafo (paralela, oblicua o aérea) tratando por tanto cualquier tipo de perspectiva cónica frente a metodologías propuestas por otros autores que siempre se han centrado en perspectivas aéreas.

Palabras clave: *Sistemas basados en agentes; reconstrucción de imágenes; perspectiva cónica; modelado basado en imágenes*

Correspondencia: Julián F. Conesa Pastor; julian.conesa@upct.es

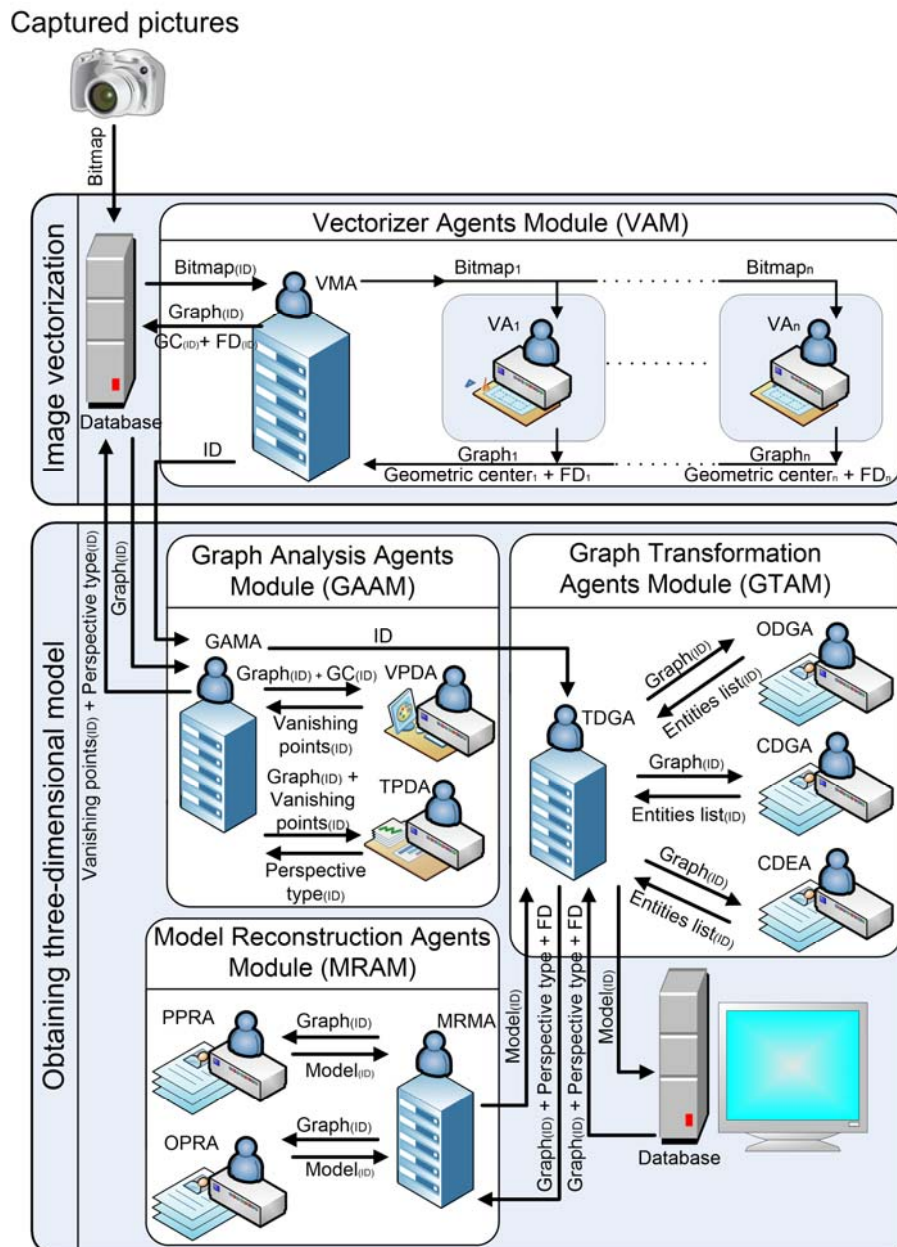


©2018 by the authors. Licensee AEIPRO, Spain. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introducción

En un trabajo previo (Conesa y Mula, 2016) presentamos una arquitectura basada en multiagentes para la reconstrucción de modelos fotografiados, es decir, para la obtención de un modelo tridimensional a partir de una fotografía. La arquitectura propuesta en ese trabajo y que se muestra en la figura 1, se centra en fotografías tomadas desde puntos de vista particulares que tras la vectorización ofrecen grafos del modelo en perspectiva cónica paralela y oblicua.

Figura 1: Diagrama funcional de la arquitectura basada en agentes



Esta arquitectura realiza la reconstrucción de modelos como sigue:

1. El proceso de reconstrucción es iniciado por el Agente Manager Vectorizador (VMA) del Módulo de Agentes Vectorizadores (VMA). Cuando el VMA lee una fotografía de la base

de datos, crea un nuevo agente vectorizador (VA) que será el encargado de vectorizar la imagen (Bitmap) para transformarla en un grafo definido por una lista de aristas y vértices. Cuando el VA finaliza su trabajo, envía el resultado al VMA quien lo almacena en la base de datos, destruye el VA y envía un mensaje al Agente Manager del Módulo de Agentes de Análisis de Grafos (GAMA).

2. Cuando el GAMA recibe un mensaje del VMA activa los agentes del Módulo de Agentes de Análisis de Grafos (GAAM) para que obtengan del grafo la información necesaria para realizar su reconstrucción. Para ello se comunica con el agente VPDA (Agente detector de puntos de fuga) quien se encarga de obtener los posibles puntos de fuga y el agente TPDA (Agente detector de tipo de perspectiva) quien se encarga de determinar a qué tipo de perspectiva corresponde el grafo y definir los puntos de fuga. Cuando el agente GAMA recibe respuesta de los agentes VPDA y TPDA, guarda la información en la base de datos y envía un mensaje al Agente de Detección de Tipología del Grafo (TDGA).
3. Para que un grafo pueda ser reconstruido por los algoritmos propuestos en este trabajo es necesario que estos correspondan a tipologías del tipo normalón o cuasi-normalón de acuerdo con la propuesta dada Dori (Dori y Tombre, 1995). Cuando el agente TDGA recibe un mensaje del agente GAMA, lee el grafo de la base de datos y determina si este corresponde a una de estas tipologías. En caso de ser así, se procede a la reconstrucción. Si no lo es, el TDGA envía mensajes a los agentes Detección de Grafos Origami (ODGA), Detección de conectividad de Grafos (CDGA) y Detección de aristas colineales (CDEA) para que modifiquen el grafo hasta conseguir esta tipología.
4. Cuando un grafo cumple con las características necesarias para ser reconstruido, el TDGA envía un mensaje al Agente Manager de Reconstrucción de Modelos (MRMA) del Módulo de Agentes de Reconstrucción de Modelos (MRAM) para que inicie la reconstrucción.
5. Cuando el MRMA recibe un mensaje del TDGA, envía un mensaje al Agente de Reconstrucción de Perspectivas Paralelas (PPRA) o al Agente de Reconstrucción de Perspectivas Oblicuas (OPRA) dependiendo del tipo de perspectiva en el que se representa el grafo y que fue analizada en el GAAM. Estos agentes realizan la reconstrucción obteniendo el modelo tridimensional y responden al mensaje devolviendo la información del modelo.
6. Cuando el MRMA recibe un mensaje del PPRA o el OPRA, responden al mensaje recibido del TDGA con el modelo reconstruido, quien finalmente lo almacena en la base de datos.

Este trabajo ha supuesto una novedosa aportación a la reconstrucción de modelos frente a otros trabajos en tanto que presenta por primera vez una arquitectura basada en agentes que permite el tratamiento en paralelo de varias imágenes y una sencilla escalabilidad a modelos de otras tipologías.

Además, no precisa de la detección de las caras del modelo a partir del grafo de partida como ocurre en otras propuestas que parten de los trabajos presentados por Marill (Marill, 1991), Leclerc y Fichler (Leclerc y Fichler, 1992) y Lipson y Shpitalni (Lipson y Shpitalni, 1996). Además, su coste computacional es mucho más bajo en comparación con otros métodos como el presentado por Liu (Liu et al., 2008) o el presentado por Lee y Fang (Lee y Fang 2012).

Una tendencia actual que persigue como objetivo final la reconstrucción se centra en la partición de modelos complejos en partes más simples, para posteriormente, una vez reconstruidas, volver a unirlas para formar un único sólido. Sin embargo, la mayoría de trabajos presentados desde este enfoque (Chen, Liu y Tang, 2007, Xue, Liu y Tang, 2010,

Liu, Chen y Tang, 2011, Yang, Liu, y Tang, 2013, Zou, Chen, Fu y Liu, 2015) necesitan como información de partida las caras definidas en el modelo. En esta línea, Fang y Lee (Fang y Lee, 2014) presentan un algoritmo de división que no precisa de la detección previa de caras, pero sus resultados no siempre son los esperados.

2. Objetivos

El trabajo descrito fue ensayado para un total de 160 grafos que corresponden a representaciones de perspectivas paralelas y oblicuas, sin embargo, la arquitectura definida no esta preparada la reconstrucción de perspectivas aéreas.

Nuestro objetivo en este trabajo ha sido la de incorporar a la arquitectura presentada la posibilidad de reconstruir modelos que estén representados en perspectiva aérea, consiguiendo de este modo en una única plataforma, un motor de reconstrucción global con independencia del tipo de perspectiva en la que este representado el grafo.

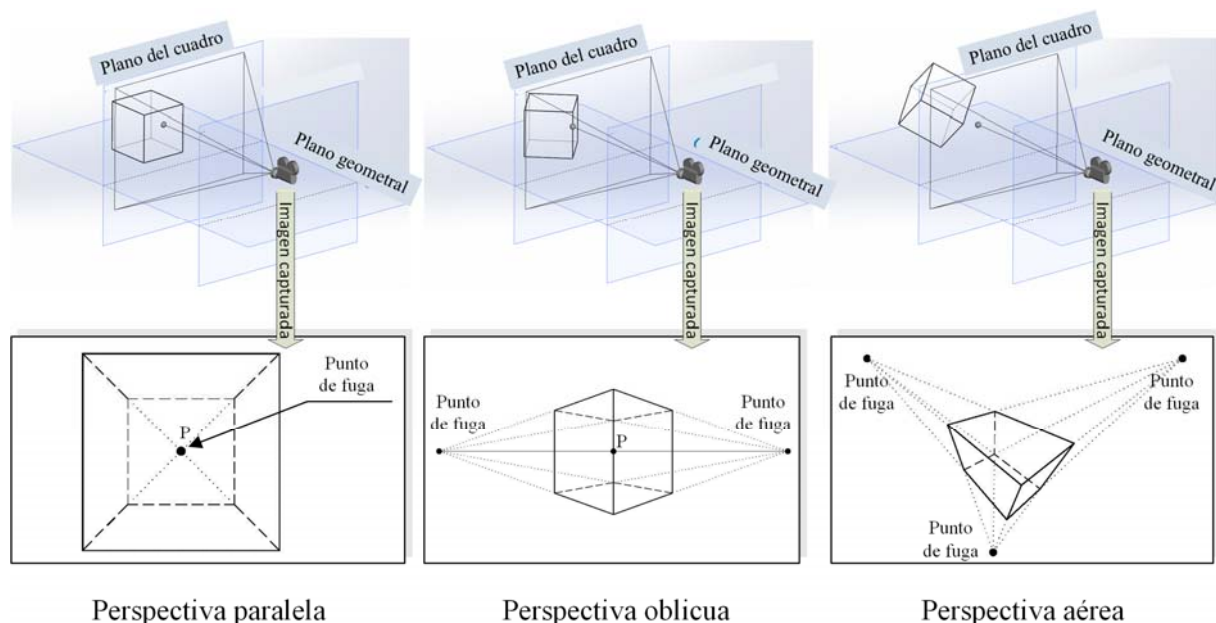
Además, se han analizado las debilidades de la arquitectura inicial y se han implementado mejoras que dan como resultado una disminución del coste computacional en el proceso de reconstrucción.

3. Metodología

3.1 Análisis del tipo de perspectiva

Nuestro primer objetivo se centra en la reconstrucción de perspectivas cónicas aéreas. Tomando como modelo la figura prismática que se representa en la figura 2, la tipología de una perspectiva cónica puede determinarse a partir de las siguientes consideraciones:

Figura 2: Tipos de perspectivas cónicas

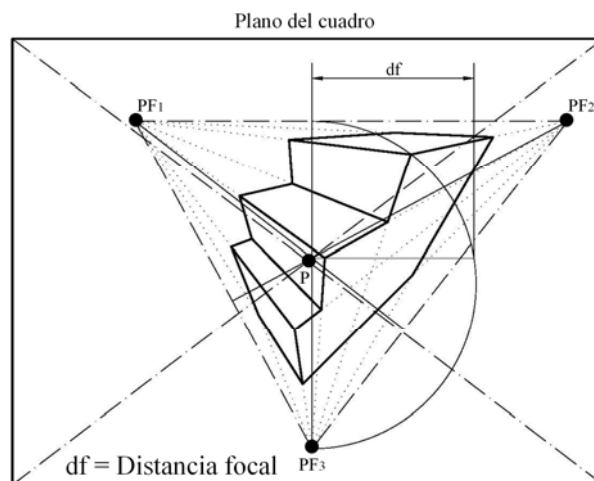


1. Una perspectiva paralela del modelo se obtiene cuando este es fotografiado con una cara apoyada en el plano geométral mientras que otra de sus caras es paralela al plano del cuadro, de tal modo que, algunas aristas del prisma quedan proyectadas según rectas horizontales o verticales y el resto concurren en un punto denominado punto de fuga que coincide con la proyección del punto vista conocido como punto principal (P).

2. Una perspectiva oblicua del modelo se obtiene cuando este es fotografiado con una cara apoyada en el plano geometral mientras que sus caras laterales son oblicuas al plano del cuadro, de tal modo que, algunas de las aristas se proyectan verticalmente y el resto concurren en dos puntos de fuga alineados horizontalmente con P.
3. Una perspectiva aérea del modelo se obtiene cuando este es fotografiado con sus caras oblicuas respecto al plano geometral y al plano del cuadro, de tal modo que, las aristas del prisma concurren en tres puntos de fuga, estando dos de ellos alineados según una horizontal y coincidiendo el ortocentro del triángulo definido por los puntos de fuga con P.

Centrado en estas características de la perspectiva cónica, el comportamiento del agente TPDA ha sido reprogramado para incluir la perspectiva cónica. Para ello, una vez detectados los posibles puntos de fuga por el agente VPDA siguiendo el algoritmo propuesto por Rother (Rother, 2002), el agente TPDA determinará que el grafo corresponde a una perspectiva cónica del tipo aérea si, existen al menos tres puntos de fuga, dos de ellos alineados según una horizontal, el ortocentro del triángulo definido por los puntos de fuga coincide con el centro del plano del cuadro y además, la distancia focal obtenida por procedimiento geométrico coincide con la distancia focal del archivo imagen.

Figura 3: Parámetros de la perspectiva cónica aérea



3.2 Reconstrucción de grafos en perspectiva aérea

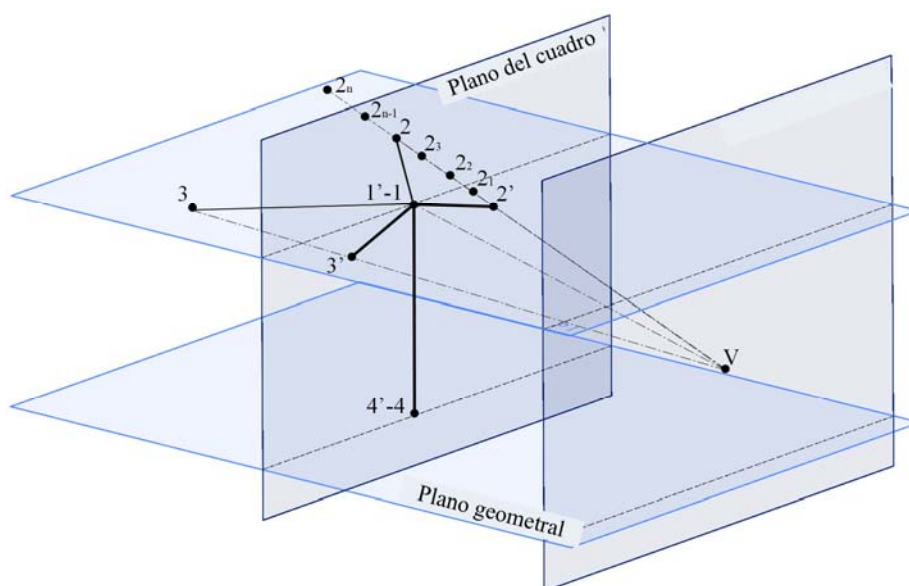
El siguiente paso al reconocimiento de perspectivas cónicas aérea y la detección de los puntos de fuga es la reconstrucción propiamente dicha, es decir, la obtención del modelo tridimensional a partir del grafo.

Como hemos visto, en la arquitectura definida, este proceso se realiza en el MRAM donde los agentes PPRA y OPRA permiten la obtención de modelos tridimensionales a partir de grafos representados en perspectivas cónicas paralelas y oblicuas. Para tratar la reconstrucción grafos representados en perspectivas aéreas hemos incorporado en el módulo un nuevo agente denominado Agente de Reconstrucción de Perspectivas Aéreas (APRA) que actuará de forma similar a los anteriores, es decir, cuando el agente MRMA recibe un mensaje del agente TDGA que contiene un grafo que representa una perspectiva cónica aérea, envía un mensaje al agente APRA, quien se encargará de realizar la reconstrucción y responderá el mensaje con las coordenadas del modelo tridimensional una vez calculadas. A continuación, se detalla el proceso ejecutado por el agente APRA para la obtención de las coordenadas tridimensionales del grafo.

El método de reconstrucción propuesto se basa en el principio de la perspectiva cónica según el cual los vértices de un modelo se encuentran sobre la recta que une el punto de vista V con la proyección de dicho vértice en el plano del cuadro. De acuerdo con esto, el vértice 2 de la figura 4, se encontrará sobre la recta trazada desde V a $2'$, lo que define un lugar geométrico de posibles soluciones ($2_1, 2_2, \dots, 2_{n-1}, 2_n$).

Considerando que las aristas que se unen en el vértice 1 son paralelas a las direcciones principales y por tanto perpendiculares entre sí, suponiendo conocidas las coordenadas de los vértices que definen una arista, por ejemplo la 1-4, podrá definirse un segundo lugar geométrico formado por el plano perpendicular a la aristas 1-4 que contendrá a todas las aristas perpendiculares a ella. De la intersección de ambos lugares geométricos es posible obtener las coordenadas del vértice 2 y de idéntica manera las coordenadas del vértice 3.

Figura 4: Obtención de coordenadas tridimensionales

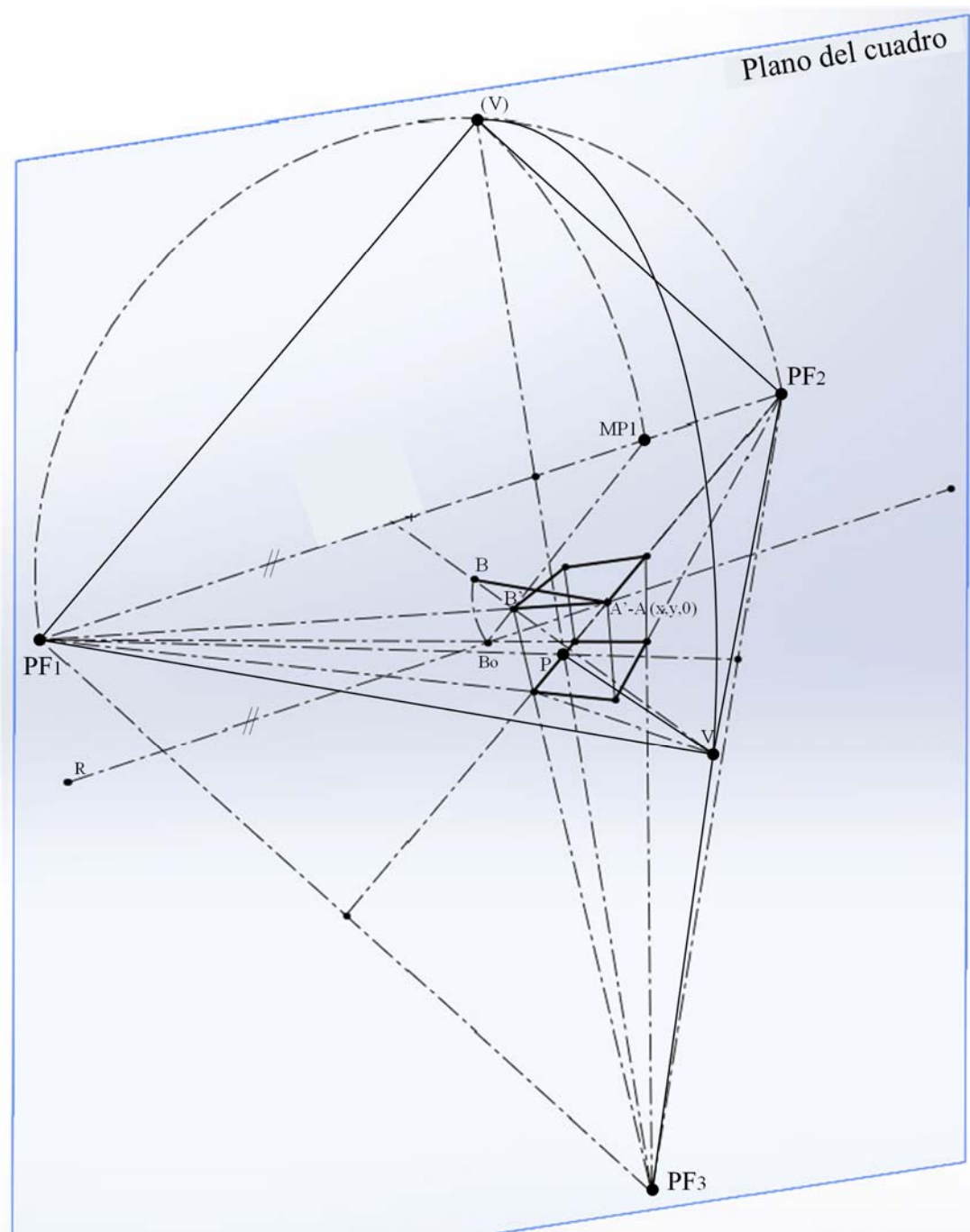


El método descrito sigue un árbol de propagación a través de vértices conectados por aristas paralelas a las direcciones principales, es decir, a través de las aristas que resulten perpendiculares en el modelo, hasta obtener las coordenadas de todos los vértices del grafo.

Nuestro objetivo se centra por tanto en este momento, en asignar coordenadas a los vértices una primera arista que permita propagar esta metodología por el resto de vértices del grafo. Para ello, hemos tenido en cuenta características de la proyección cónica aérea según las cuales, (ver figura 5) podemos asegurar que, dada la perspectiva aérea de un modelo desde un punto de vista V , las proyecciones de las aristas orientadas según las direcciones principales convergerán en tres puntos de fuga (PF_1, PF_2, PF_3) que definen un triángulo cuyo ortocentro coincide con la proyección del punto de vista (P).

Abatiendo cualquiera de las caras de la pirámide definida por los puntos de fuga y el vértice V sobre el plano del cuadro (en nuestro caso la cara definida por los vértices V, PF_1 y PF_2) es posible obtener los puntos métricos de cada punto de fuga llevando la distancia de V al punto de fuga sobre el eje utilizado para el abatimiento (MP_1). Los puntos métricos permiten determinar la verdadera magnitud de segmentos de línea que convergen sobre los distintos puntos de fuga.

Figura 5: Asignación de coordenadas a una arista del grafo



Basados en estos principios hemos definido el algoritmo a ejecutar por el APRA para determinar las coordenadas de un modelo de acuerdo con los siguientes pasos:

1. Asignar coordenadas iniciales a un vértice arbitrario de la proyección, A en nuestro caso, manteniendo sus coordenadas (x,y) iniciales y haciendo $z=0$.
2. Determinar las coordenadas de una arista que pase por dicho vértice, arista AB en nuestro ejemplo, de acuerdo a los siguientes puntos:
 - Dado que la proyección $A'B'$ fuga hacia PF_1 , trazamos por A' una paralela a la recta definida por PF_1 y PF_2 .

- Unimos el punto métrico MP1 con la proyección del vértice B' (dado que MP1 es el punto métrico de PF_1) y obtenemos el punto de corte B_0 sobre la paralela trazada en el punto anterior. En $A'B_0$ tendremos la verdadera magnitud del segmento AB.
 - Obtenemos las coordenadas del vértice B como el punto sobre la recta VB que diste una distancia $A'B_0$ de A.
3. Obtenidas las coordenadas de una arista del modelo, se ejecuta la metodología de propagación anteriormente expuesta.

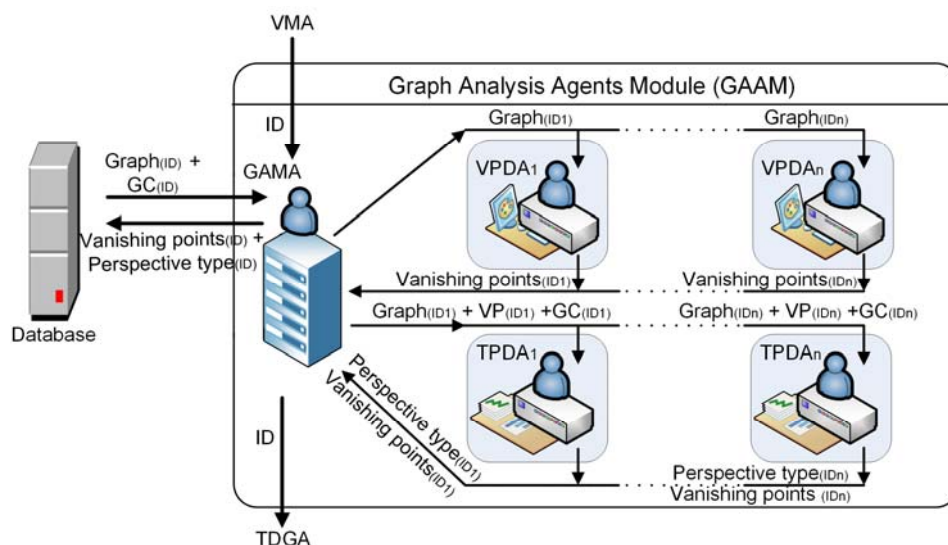
3.3 Mejoras implementadas en la arquitectura.

Analizados los costes computacionales de cada uno de los agentes que intervienen en la arquitectura, se ha podido comprobar que, debido a las operaciones que deben realizar los agentes del GAMA y especialmente desde que se ha incorporado el análisis de grafos en perspectiva aérea, su tiempo de ejecución es muy alto en comparación con el resto de agentes, por lo que en ocasiones los agentes del GTAM y MRAM quedan a la espera de recibir nuevos mensajes haciendo que el coste computacional global de arquitectura aumente.

Por ese motivo, la arquitectura de este módulo ha sido modificada pasando de una estructura estática a una estructura dinámica en la que el agente GAMA coordina una lista de agentes VPDA y TPDA que son creados y destruidos en tiempo de ejecución de forma similar a como se realiza en el VAM. La nueva estructura definida para el GAAM y que se muestra en la figura 6 actúa del siguiente modo:

1. Cuando el GAMA recibe un mensaje del VMA, crea un nuevo VPDA y le envía un mensaje que contiene como información el grafo a analizar. El nuevo VPDA obtiene los posibles puntos de fuga del grafo y responde al mensaje recibido del GAMA con sus coordenadas.
2. Cuando el GAMA recibe respuesta de un VPDA, crea un nuevo TPDA al que envía un mensaje que contiene el grafo y la información facilitada por el VPDA. Posteriormente destruye el VPDA. El nuevo TPDA será el encargado de determinar el tipo de perspectiva que representa el grafo y los puntos de fuga que deben ser considerados. Cuando el TPDA responde al mensaje enviado por el GAMA con esta información, el GAMA la guarda en la base de datos, destruye el TPDA y envía un mensaje al TDGA.

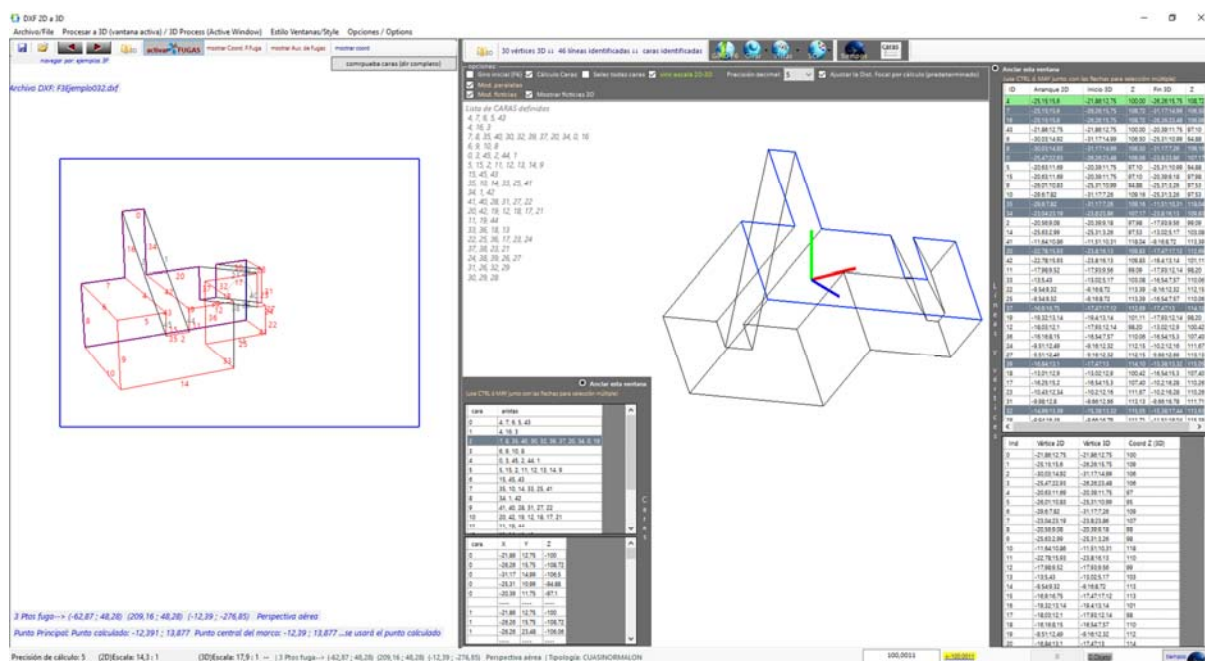
Figura 6: Nueva estructura del GAAM



4. Resultados

La arquitectura basada en agentes propuesta en este trabajo ha sido implementada usando la plataforma JADE bajo Eclipse. El entorno de la aplicación que se muestra en la figura 7 esta dividido en dos ventanas. En la ventana izquierda se muestra el grafo de partida y permite visualizar los puntos de fuga detectados, las direcciones principales, la tipología del modelo y el tipo de perspectiva detectado. En la ventana de la derecha se muestra el modelo obtenido tras la reconstrucción y una serie de menús desplegables que contienen los vértices del modelo, sus aristas, y las caras. Interactuando en los menús desplegables es posible la ejecución paso a paso para visualizar el árbol de propagación del método de reconstrucción.

Figura 7: Entorno de la aplicación

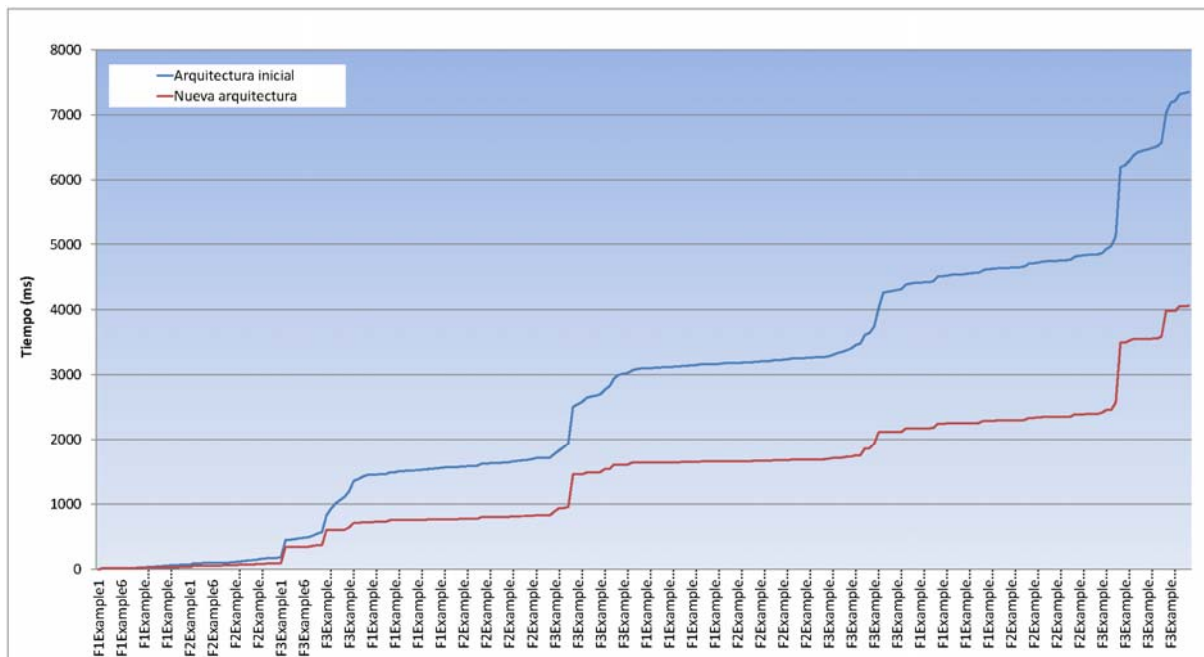


Para evaluar los métodos propuestos se han ensayado, además de los 160 grafos que representan perspectivas paralelas y oblicuas propuestos por Conesa y Mula (Conesa, 2016) otros 80 grafos que se corresponde con los mismos modelos de la publicación anterior, pero ahora representados en perspectiva aérea. El total de 240 grafos han sido reconstruidos con éxito.

En lo referente al análisis de coste computacional, en la figura 8 se muestra una comparativa entre el coste de la arquitectura inicial de Conesa y Mula (Conesa, 2016) y la nueva arquitectura propuesta cuando se envían para la reconstrucción los 240 modelos ensayados.

Como puede observarse la disminución del coste computacional puede considerarse notable al poco tiempo de iniciarse la reconstrucción de los modelos, si bien, como era de esperar, los mayores beneficios de la arquitectura dinámica definida para el módulo GAAM se obtienen al aumentar el número de modelos a reconstruir.

Figura 8: Comparación de costes



5. Conclusiones

En este trabajo se ha presentado una propuesta para la reconstrucción de modelos a partir de fotografías con independencia del punto de vista del que se realizar la fotografía al agrupar en un único motor de reconstrucción las perspectivas cónicas paralelas, oblicuas y aéreas.

El sistema se ha implementado por primera vez en una arquitectura basada en múltiples agentes que permiten ejecutar paralelamente los distintos algoritmos necesarios actuando de forma automática a partir de un conjunto de decisiones basadas en la tipología de la perspectiva y del grafo previamente analizada. La arquitectura se ha estructurado en cuatro módulos: i) un módulo para la vectorización de las imágenes, ii) un módulo que detecta puntos de fuga y determina la tipología de la perspectiva, iii) un módulo que determina la tipología de los grafos y los modifica en caso necesario para que puedan ser reconstruidos, y iv) un módulo que realiza la reconstrucción propiamente dicha. Además, todo el proceso se realizar de forma completamente automática sin necesidad de interactuar con el usuario.

Con el fin de evaluar la arquitectura propuesta, se han realizado un total de 240 ensayos sobre grafos que representan superficies planas y modelos que contienen o no representaciones de aristas ocultas. Todos ellos representan modelos de tipología normalón o cuasi-normalón, logrando un porcentaje de éxito del 100%.

La arquitectura presentada, modulada en agentes, permite con un mínimo esfuerzo incorporar nuevos algoritmos que permitan reconstruir modelos de otras tipologías, constituyendo por tanto una herramienta escalable para la reconstrucción de nuevos modelos.

Referencias

- Chen, Y., Liu, J. & Tang, X. (2007). A divide-and-conquer approach to 3D object reconstruction from line drawings. *11th IEEE International Conference on Computer Vision*, 1807-1814.
- Conesa, J. & Mula, F. (2016). An agent-based paradigm for the reconstruction of conical perspectives. *Expert Systems With Applications* (55), 1-20, DOI information: 10.1016/j.eswa.2016.02.003
- Dori, D., & Tombre, K. (1995). From engineering drawings to 3D CAD models: are we ready now?. *Computer Aided Design*, 27(4), 243- 254.
- Fang, F. and Lee, Y.T. (2014). Efficient decomposition of line drawings of connected manifolds without face identification. *Computer Aided Design*, Vol. 51, 18-30.
- Leclerc Y. & Fischler M. (1992). An optimization –based approach to the interpretation of single line drawings as 3D wire frames. *International Journal of Computer Vision*, 9(2), 113-136.
- Lee, Y.T. & Fang, F. (2012) A new hybrid method for 3D object recovery from 2D drawings and its validation against the cubic corner method and the optimisation-based method. *Computer-Aided Design*, 44(11), 1090-1102.
- Lipson, H & Shpitalni, M. (1996). Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer Aided Design*, 28(8), 651-663.
- Liu, J.Z., Cao, L.L., Li, Z.G. and Tang, X.O. (2008) Plane-based optimization for 3D object reconstruction from single line drawings. *IEEE Transactions on pattern analysis and machine intelligence*. Vol. 30(2), 315-327.
- Liu, J., Chen, Y. & Tang, X. (2011). Decomposition of Complex Line Drawings with Hidden Lines for 3D Planar-Faced Manifold Object Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 33(1), 3-15.
- Marill, T. (1991). Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects. *International Journal of Computer Vision*, 6(2), pp. 147-161.
- Rother, C. (2002). A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10), 647–55.
- Xue, T., Liu, J. & Tang, X. (2010). Object Cut: Complex 3D Object Reconstruction Through Line Drawing Separation. *23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1149-1156.
- Yang, L., Liu, J. & Tang, X. (2013). Complex 3D General Object Reconstruction from Line Drawings. *IEEE International Conference on Computer Vision (ICCV)*, 1433-1440.
- Zou, C.Q., Chen, S.F., Fu, H.B. & Liu, J.Z. Progressive 3D Reconstruction of Planar-Faced Manifold Objects with DRF-Based Line Drawing Decomposition. *IEEE Transactions on visualization and computer graphics*. Vol. 21(2), 252-263, DOI:10.1109/TVCG.2014.2354039.