METODOLOGÍA Y HERRAMIENTAS DE PLANIFICACIÓN ESTÁTICA PARA APLICACIONES DE VISIÓN ARTIFICIAL

F. A. Candelas, S.T. Puente, F. Torres
Grupo de Automática y Visión Artificial (http://www.disc.ua.es/gava/),
Dpto. Física, Ingeniería de Sistemas y Teoría de la Señal,
Universidad de Alicante.
fcandela@disc.ua.es, spuente@disc.ua.es, medina@disc.ua.es

Resumen

Se presenta a continuación un nuevo entorno de trabajo para la especificación y simulación mediante esquemas gráficos de algoritmos de visión artificial de tiempo real, y que, como principal novedad, permite evaluar distintos métodos de planificación y asignación de tareas para los esquemas diseñados. El entorno se compone de varios módulos, lo cual le dota de gran flexibilidad.

Palabras Clave: planificación, simulación, visión artificial, tiempo real.

1 INTRODUCCIÓN

El entorno de simulación y planificación se compone de cuatro módulos, tres de ellos aplicaciones ejecutables, más una base de datos. La figura 1 refleja estos módulos y los principales flujos de información entre ellos y con el usuario.

La estructura distribuida de las aplicaciones proporciona un entorno flexible. El entorno visual y el Interprete se comunican mediante socktes TCP/IP dentro de una red local (o en la misma máquina). El grafo de tareas elementales se pasa al Planificador como un archivo.

Las herramientas de simulación, el entorno visual y el interprete, comenzaron a desarrollarse dentro del proyecto CICYT coordinado "Construcción de un Entorno para la Investigación y Desarrollo en Visión Artificial" (TAP96-0629-CO4) [2][8]. El planificador y la base de datos son extensiones no incluidas en proyecto original que suponen una importante mejora.

Seguidamente, en el punto 2 de este artículo se describen las herramientas implicadas en la simulación. El punto 3 trata sobre la base de datos y expone la información que mantiene ésta. Finalmente, la herramienta de planificación se describe en el punto 4 junto con el proceso llevado a cabo para obtener el grafo de tareas elementales.

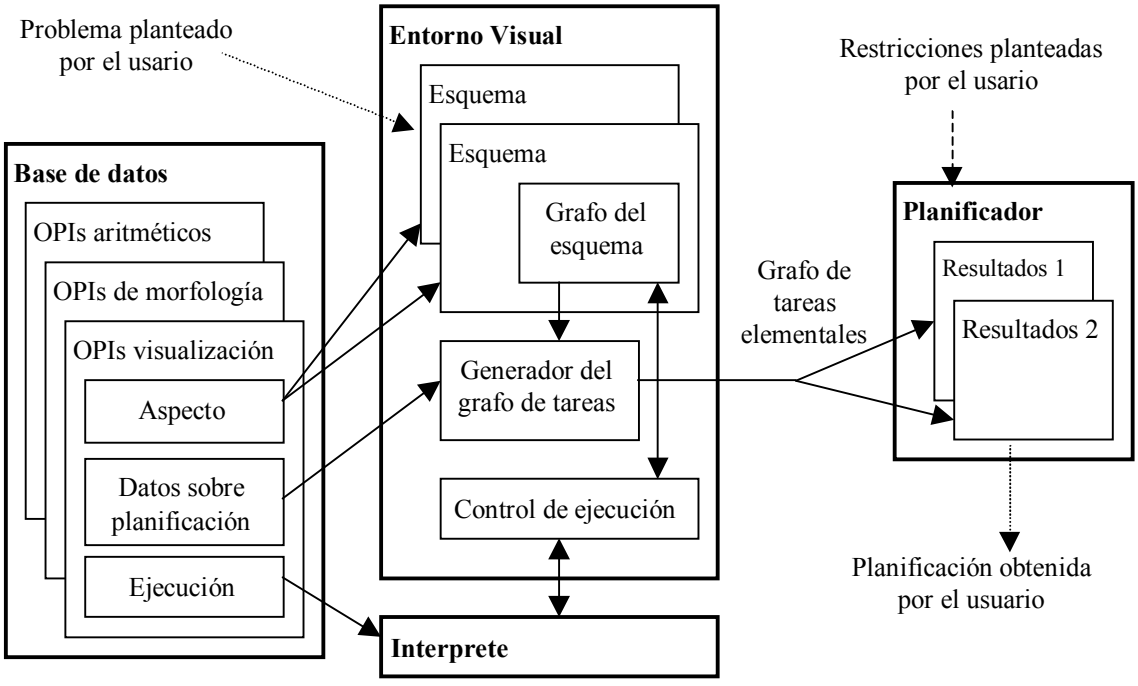

Figura 1: Módulos del entorno

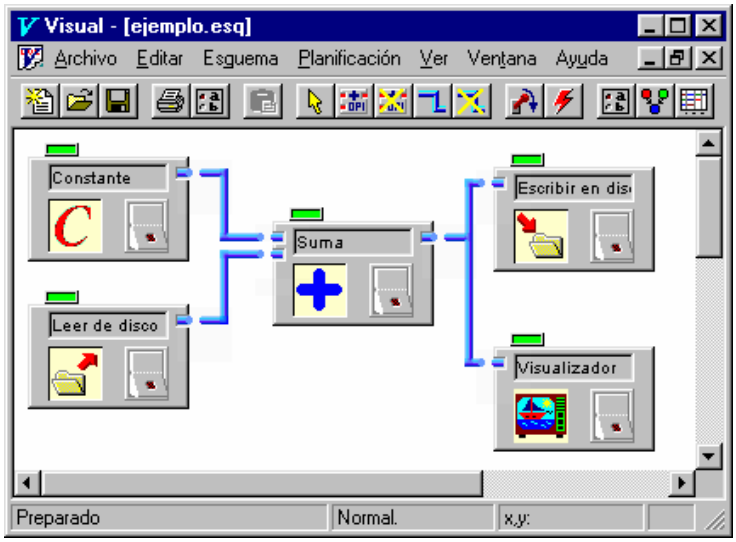

Figura 2: ejemplo de esquema en el entorno visual

2 ESPECIFICACIÓN Y SIMULACIÓN

2.1 EL ENTORNO VISUAL

Con el entorno visual el usuario puede especificar un algoritmo de procesamiento de imágenes como un esquema gráfico compuesto por OPIs (Objetos de Procesamiento de Imágenes). Estos se pueden conectar entre sí mediante unas tuberías que representan el intercambio de datos o imágenes. (figura 2). El aspecto de los OPIs y de las tuberías refleja claramente el estado de ejecución del esquema, ofreciendo el entorno visual un aspecto más ilustrativo que otros programas existentes [2][3][8].

Cada OPI representa una operación a realizar, y posee una serie de entradas a donde llegarán las imágenes que debe operar procedentes de otros OPIS, y de salidas por las que devuelve los resultados tras su ejecución. Las tuberías permiten conectar unas salidas con una o más entradas. Los posibles OPIs se clasifican en distintos tipos según la clase de operaciones que realizan [2].

Un OPI tiene un conjunto de propiedades según su tipo que permiten definir los parámetros de la operación a realizar (operación, número de entradas, factores que influyen en la operación, nombres de archivos, etc.).

2.2 EL INTERPRETE

Es el encargado de ejecutar el código asociado a cada OPI. El hecho de que se trate de una aplicación independiente del entorno visual facilita la modificación o actualización de su biblioteca de funciones de procesamiento. Además así resulta posible disponer de diferentes interpretes para distintos objetivos; simulación en docencia, trabajo sobre hardware de adquisición o procesamiento con bibliotecas propias, interprete programado sobre una máquina con otro S.O., etc.

Básicamente el interprete espera a recibir del entorno visual mensajes con el identificador de OPI a ejecutar, y referencias sobre datos que se deben operar (figura 3). Luego, accediendo a la base de datos determina el código asociado a dicho OPI para ejecutarlo. El interprete gestiona también la asignación de memoria para los datos o imágenes. Cuando el código de un OPI se ha ejecutado se devuelve un mensaje al entorno visual que referencia ese OPI y las variables donde se han depositado los resultados. La secuencia de ejecución de los OPIs de un esquema es determinada por el entorno visual conforme a las conexiones de ese esquema, y no por el interprete.

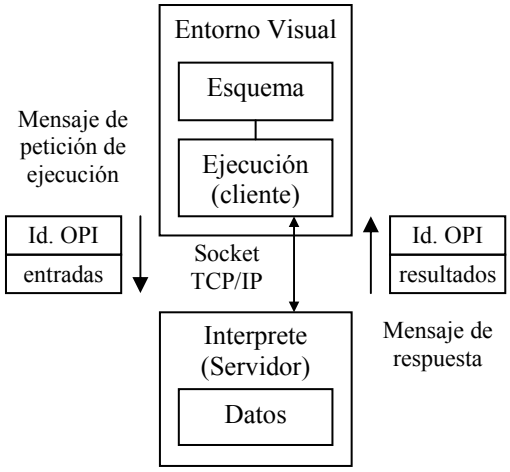

Figura 3: comunicación entorno visual – interprete

Este modo de trabajo permite plantear un interprete que pueda ejecutar OPIs en paralelo, ya que el entorno visual puede generar diversas peticiones de

ejecución para OPIs cuyas entradas sean independientes antes de recibir las respuestas del interprete.

3 LA BASE DE DATOS

La base de datos es una tabla Access con un registro por cada posible tipo de OPI [3]. Cada registro contiene información sobre cuatro aspectos:

a) General. Identificadores de tipo de OPI y de operación que realiza, además de descripción, nombre, etc. Los identificadores indexan el registro.

b) Datos sobre planificación. Se especifica el subgrafo de tareas elementales que ejecuta el OPI, así como un indicador de si el OPI es divisible, es decir, se puede dividir su ejecución en varios flujos de tareas paralelos de menor tamaño. En caso de ser divisible se especifica también la máxima capacidad de tratamiento de imágenes del subgrafo. Cada tarea del subgrafo tiene indicado su tipo (cpu o tapi), y sus costes.

c) Aspecto del OPI. Aspecto del dialogo de configuración del OPI, el icono asociado, o datos como el número de entradas y salidas.

d) Características de ejecución. Como son el texto del código a ejecutar y los tipos de variables internas.

El hecho de disponer de la información sobre los OPIs y sus características en la base de datos facilita la especificación de nuevos OPIs o la modificación de estos de forma dinámica, sin necesidad de recompilar la aplicación para actualizar su código. Esta información se carga al iniciar el entorno de manera que se disponga de los nuevos OPIs al realizar nuevos esquemas. Sin embargo presenta problemas al intentar cargar esquemas viejos realizados con OPIs que ya no existen en la Base de Datos actual o cuyas características se han cambiado. Estos problemas no son significativos ya que lo habitual es especificar nuevos OPIs, no eliminar o cambiar los existentes.

Por otra parte la información sobre las tareas de cada OPI es necesaria para generar el grafo de tareas elementales de un esquema. Este grafo sirve como entrada al planificador de tareas.

4 PLANIFICACIÓN DE TAREAS

4.1 EL PLANIFICADOR

La aplicación del Planificador de Tareas constituye el último paso en el diseño de sistemas de visión artificial [4][5]. Dependiendo del hardware disponible se debe calcular una planificación espacial y temporal optima.

El planificador diseñado es genérico y realista: un planificador estático que considera interrupción de tareas y tiene en cuenta las relaciones de precedencia [6]. De este modo, se puede asegurar que ofrecerá una solución factible y posible de llevar la práctica [7].

Uno de los principales objetivos de este planificador es reducir el tiempo final de ejecución de un grupo de subtareas, a la vez que se obtiene información sobre la posibilidad de ejecutarlas dentro de ciertos límites de tiempo [7].

La mayor ventaja de la planificación estática [1] es que el tiempo de cálculo requerido para la planificación se incluye en las tareas de compilación de un programa, lo que resulta más eficiente que realizar una planificación dinámica en el tiempo de ejecución. Sin embargo, no resulta nada fácil considerar eventos externos que pudieran aparecer durante la ejecución del sistema.

Se ha optado por la planificación estática porque esta encaja perfectamente en la naturaleza de los sistemas tratados: sistemas de procesado de visión artificial en los que se conoce de antemano las tareas a ejecutar o el hardware disponible, y en los que no se producen eventos inesperados. Aún más, en esos sistemas el usuario necesita conocer de un modo fiable los tiempos de ejecución asociados a un cierto esquema, y estos datos no los puede proporcionar una planificación dinámica [2][7].

Para realizar la planificación se requiere conocer el grafo de subtareas en que se divide el sistema especificado en un esquema especificado en el entorno visual. Este grafo se genera en el entorno visual siguiendo el procedimiento descrito en el punto 4.2 [3].

Cada subtarea del grafo puede pertenecer a uno de los siguientes tres tipos: cpu, tapi y cpu/tapi. Las primeras son tareas a ejecutar un procesador genérico o CPU del sistema destino. Las segundas representan operaciones de bajo nivel a ejecutar en tarjetas de Adquisición y Procesamiento de Imágenes o TAPIs. Las terceras se deben ejecutar en ambos tipos de procesadores y representan un intercambio de información entre estos [2][3][7].

Las tareas cpu o tapi pueden ser interrumpibles o no interrumpibles. Las cpu/tapi se consideran no interrumpibles. Cada subtarea tiene asignado un coste de ejecución, un coste de ejecución límite para ser realizada (deadline) y, opcionalmente, unos costes de interrupción y de planificación para tareas

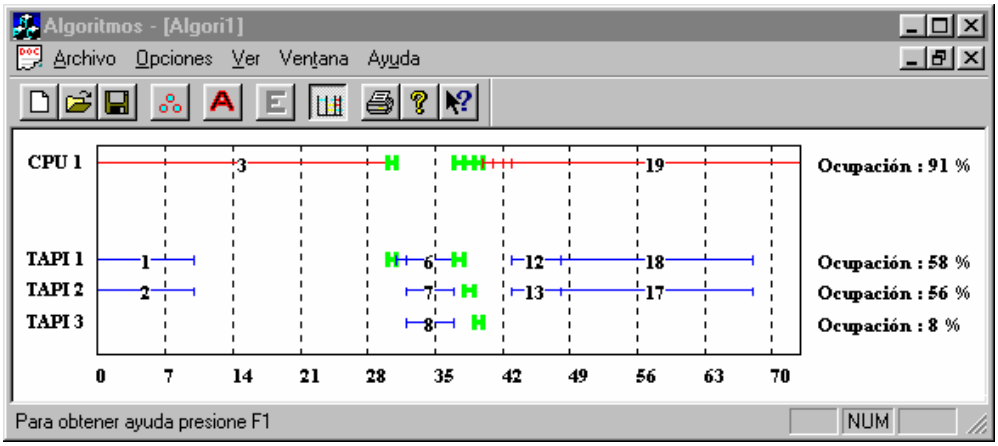

Figura 4: ventana resultado en el planificador

interrumpibles. También, se consideran las relaciones de precedencia entre tareas, a las que se les puede asignar un coste, para representar, por ejemplo, un coste de transferencia de datos [7].

El planificador desarrollado permite múltiples posibilidades de algoritmos de asignación y planificación de tareas. También permite especificar el número de procesadores disponibles, u obtener el número de procesadores máximo número de procesadores necesario para una ejecución del menor tiempo posible. El usuario puede elegir entre esas posibilidades de acuerdo a los resultados que desea obtener [2][7].

La figura 4 muestra el resultado generado por el planificador para el esquema de la figura 2 considerando un sistema destino con una CPU y tres TAPIs y una planificación espacio-temporal con interrupción de tareas. Los pequeños intervalos de tiempo con trazo grueso entre los instantes 28 y 42 representan las tareas cpu/tapi.

4.2 GENERACIÓN DEL GRAFO DE TAREAS

Dentro del entorno visual existe la opción de generar el grafo de tareas elementales al ejecutar un esquema, haciendo uso de la información de la base de datos. Este proceso se subdivide en tres fases bien diferenciadas.

a) Inicialización. Al ejecutar un primer OPI del esquema se inicia la generación del grafo global de tareas elementales. Se abre la conexión con la base de datos para su utilización, así como se cargan en memoria las parejas de tareas cpu/tapi-tapi y cpu/tapi-cpu que servirán para conectar las tareas cpu con las tapi si las unas necesitan información de las otras.

b) Generación de tareas. Para cada OPI del esquema que se ejecuta, en primer lugar se averigua todos los OPIs alcanzables desde él, es decir, los que requieren sus datos de salida. Además se determina el tipo de operación que realiza el OPI así como la función a la que corresponde dentro de la clase y el tamaño medio de las imágenes que recibe.

Partiendo de la información del tipo de operación u de la función específica se busca en la base de datos la información relativa a la planificación del OPI, entre la que destaca el subgrafo de tareas de éste.

Con la información obtenida se comprueba si el subgrafo es divisible o no. Para ello se compara el tamaño de la imagen recibida con el máximo tamaño de imagen permitido para la operación especificado en la base de datos.

En el caso de ser divisible se procede a calcular en cuantas partes es divisible la operación del OPI. Por cada división, la operación tendrá asociada una copia del subgrafo. Con esta información se calcula el intervalo de actuación en [0-1] que corresponde a cada subgrafo que hay que generar, teniendo en cuenta que la distribución se realizará en igualdad de condiciones para todos los subgrafos.

Una vez conocido el identificador de subgrafo que hay que colocar en el grafo global y los intervalos de actuación, se añaden al global cada copia del subgrafo. Concluido esto, hay que realizar la conexión con las tareas precedentes, y para ello hay que añadir aristas en el grafo global teniendo en cuenta de que tareas es sucesora la actual, el tipo de las tareas precedentes (cpu o tapi) y el tipo de la actual. Así se determina si es necesario introducir un conjunto de tareas cpu/tapi-cpu o cpu/tapi-tapi. Además si la tarea es divisible se tiene en cuenta el factor de actuación, de tal manera que sólo se conectará con aquellos predecesores cuya intersección del intervalo de actuación con el propio no sea nula.

Cuando se han añadido las aristas necesarias para el subgrafo entonces se añade a una lista de sucesores los hijos del OPI en proceso con un factor de

actuación igual al del subgrafo actual. Esta lista permitirá conocer los intervalos de actuación en una próxima ejecución de esta fase.

c) Finalización. Esta fase se realiza cuando ya se ha realizado la fase 2 para todos los OPIs del esquema de trabajo. En ella se almacena en disco el resultado, el grafo global, y se cierra la conexión con la base de datos. Así queda ya disponible el grafo para ser utilizado por el planificador.

La figura 5 muestra una ventana del planificador en donde se dibuja el grafo generado a partir del esquema de la figura 2.

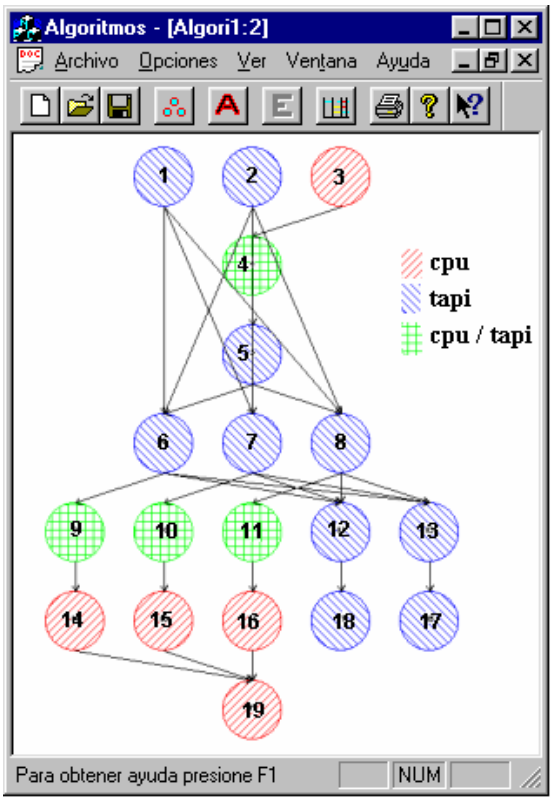

Figura 5: grafo de tareas

5 CONCLUSIONES

El entorno de especificación y el planificador se pueden utilizar con diferentes interpretes, ya que las aplicaciones son independientes. Además se puede adaptar el interprete sin alterar el resto de módulos. Por otra parte, el hecho de que las propiedades de los OPIs estén almacena en una base de datos permite la actualización de estos sin recompilar las aplicaciones. Bajo estas consideraciones se puede afirmar que el entorno aporta una ventajosa flexibilidad.

Debido al aspecto gráfico e ilustrativo de los esquemas, el entorno resulta adecuado para docencia sobre procesamiento de imágenes o visión artificial. Además, la novedosa opción de realizar una planificación de las tareas de los esquemas extiende su utilización a aplicaciones de visión artificial de tiempo real profesionales.

Agradecimientos

Este trabajo es una continuación no subvencionada del proyecto CICYT TAP96-0629-CO4-01.

Referencias

[1] Behrooz A. Shirazi, Ali R. Hurson, Krisha M. Korvi, (1995) “Scheduling and Local Balancing in Parallel and Distributed Systems”, IEEE Press.

[2] F.A.Candelas, (1998) "Simulación y Planificación en Entorno Distribuido Para Algoritmos en Tiempo Real de Visión Artificial", Memoria de Investigación, Universidad de Alicante.

[3] F.Torres, F.A.Candelas, S.T.Puente, L.M.Jiménez, C.Fernández, R.J.Agulló, (1999) "Simulation and Scheduling of Real-Time Computer Vision Algorithms", Lecture Notes In Computer Science - Computer Vision Systems, ICVS’99 Las Palmas, vol 1542, Ed. Springer, pp. 98-114.

[4] J.M.Sebastián, F.Torres, O.Reinoso, J.L.Bello, E.Barroso, (1995) "Parallel Processing and Scheduling Techniques Applied to the Quality Control of Bill Sheets", Proc. 12 IAPR (IEEE), pp. 399-403.

[5] J.M. Sebastián, F.Torres, R.Aracil, O.Reinoso, L.M.Jiménez, D.García, (1997) "Job-Shop Scheduling applied to Computer Vision", Proc. SPIE, vol. 3166, pp. 158-169.

[6] J. Xu, D.L. Parnas, (Enero 1993) “On Satisfying Timing Constrains in Hard-Real-Time Systems”, IEEE Transactions of Software Engineering, 19-1.

[7] S.T.Puente, (1998) "Asignación de Tareas y Planificación en Entornos Multiprocesador Reales. Aplicación a un Sistema de Inspección Visual", Proyecto Fin de Carrera, Universidad de Alicante.

[8] (1998) "Informe final del proyecto CICYT TAP96-0629-CO4; Construcción de un Entorno para la Investigación y Desarrollo en Visión Artificial".