Sistema de reconstrucción estéreo en tiempo real y su evaluación con ruido.

Luke, J.P.[(1)], Rosa, F.[(1)], Pérez Nava, F.[(2)], Marichal-Hernández, J.G.[(1)] y Rodríguez-Ramos, J.M.[(1)]
jpluke@ull.es, frosa@ull.es, fdoperez@ull.es, jmariher@ull.es y jmramos@ull.es
[(1)]Dpto. de Física Fund. y Exp, Electrónica y Sistemas. Universidad de La Laguna. Av. Astrofísico F^{co} Sánchez s/n.
[(2)]Dpto. de Estadísta, Investigación Operativa y Computación. Universidad de La Laguna.

***Abstract*—This paper depicts an integrated system for evaluate the stereo reconstruction of a stream of virtual stereo scenes with noise. We use an inmersive virtual scene system to produce a simulated stereo par with noise and its corresponding distances. From this stream we reconstruct the distance map using a Graphic Processing Unit with a message passing energy minimization algorithm in real time up to 20 frames per second with size from 32x32 to 100x100. We are able to evaluate the accuracy of the system comparing the two distance maps. System has been prepared to adquire from real cameras and produce dense distance maps of aproximate 5 centimeters depth resolution. The chosen reconstruction technique was a TRW algorithm for three-dimensional reconstruction which was implemented on a Graphics Processing Unit (GPU), for accelerating the performance and making real time reconstuction and accuracy estimation while navigating posible.**

I. Introducción

Hacer que un ordenador vea es algo a lo que los expertos en los años sesenta le atribuían una dificultad de proyecto de fin de carrera. Casi medio siglo después sigue sin resolver y parece una tarea formidable. Durante la última década se ha producido un ingente desarrollo y comprensión de la geometría de la visión con múltiples vistas y se han resuelto problemas que se pensaban irresolubles entonces. Uno de los logros obtenidos es establecer correspondencia entre dos imágenes dadas, sin más información que las dos imágenes y generar la posición de los puntos en 3D y las cámaras que produjeron esas imágenes [1]. El objetivo de la reconstrucción estéreo de escenas 3D es algo más complicado ya que se necesita establecer relaciones entre los puntos recuperados.
El problema en el sentido contrario es lo que en gráficos por ordenador se conoce como "*renderizado*" y que se entiende como el problema directo, en el que, dado un conjunto de cámaras, sus posiciones y una escena con iluminación y materiales, calculamos las imágenes que verían esas cámaras. Ambos problemas tienen sus propias dificultades, aunque generalmente se considera más difícil el problema inverso de reconstrucción.

El desarrollo de técnicas eficientes de minimización de la energía han jugado un papel fundamental en el problema de reconstrucción estéreo. Tres de estas técnicas son: –el recorte de grafos, "graph cut", –la propagación de esperanza, "belief propagation", (BP) y una introducida más recientemente –el paso de mensajes para balance del arbol, "Tree-reweighted message passing", (TRW) [2]. Este último ha sido comparado con los anteriores mostrando una mejor habilidad en encontrar el mínimo óptimo en algunas condiciones, pero disminuyendo su velocidad de convergencia al aumentar la conectividad del problema [3], [4], lo que hace más difícil su utilización en tiempo real (> 15 frames/s), asunto que abordamos en este trabajo.

Por otro lado, el hardware para la generación de gráficos 3d está diseñado para resolver el problema directo, sin embargo, más recientemente se ha abordado el problema inverso, aplicando las unidades de procesamiento gráfico (GPU) a problemas de propósito general (GPGPU) [5]. Usadas para la reconstrucción estéreo abren la posibilidad de procesamiento en tiempo real utilizando algoritmos de gran complejidad como el TRW.

En este trabajo se utiliza un lazo cerrado que permite evaluar la calidad de la solución inversa obtenida en presencia de ruido de cámara.

II. Diseño del sistema y reconstrucción

El sistema está basado en un trabajo anterior en lazo abierto [6] y consta de varias partes. En la figura 1 se muestra el esquema general de funcionamiento del sistema. Una de ellas es el simulador inmersivo de distancias en escenas virtuales (SIDEV), que es un módulo que permite la generación de pares estéreo correspondientes a una escena con unos parámetros de cámara conocidos. Los datos obtenidos del SIDEV se pasan al módulo recuperador estéreo, que en este caso emplea el algoritmo TRW, obteniéndose un mapa de distancias a partir de la escena. Este mapa de distancias se compara con el mapa de distancias verdaderas obtenido con el SIDEV, dando como resultado una estimación de la calidad del algoritmo usando un conjunto de métricas de calidad.

A. Simulador inmersivo de distancias en escenas virtuales (SIDEV)

El simulador inmersivo de distancias en escenas virtuales (SIDEV) permite la síntesis de imagen mientras simula la navegación de una cámara estéreo en una escena virtual, así como el mapa de distancias correspondientes a cada par estéreo generado. A estos datos se añaden los parámetros de la cámara con el fin de que puedan ser utilizados posteriormente por el algoritmo de recuperación
Antes de introducir al usuario en el mundo virtual es necesario disponer de una descripción de ese mundo. Para esto se ha elegido el lenguaje VRML que se mantiene como estandar, y que permite importar escenas creadas en plataformas de diseño 3D como AutoCAD, 3DStudio, Blender, etc. Una vez que se dispone de la descripción de la escena en VRML, el cargador de escenas (CE) la analiza para crear las estructuras

de memoria necesarias para su posterior renderizado. Este paso de renderizado requiere registrar la posición del usuario en todo momento, ajustar la escala de medida del mundo simulado, y posicionar una cámara, con unas características adecuadas (control de cámara (CC)), en ese mundo simulado. Los movimientos del usuario son registrados por el control de navegación (CN) y se pasan al módulo de renderizado (R) para que proyecte el par de imágenes estéreo correspondientes al punto de vista en cada momento. Esto se consigue mediante el estándar OpenGL. Además de la pareja de imágenes estéreo se genera el mapa de distancias verdaderas (DV) correspondiente al mismo, lo que resulta de vital importancia para cerrar el lazo de evaluación de algoritmos de recuperación estéreo que se presenta en este trabajo.

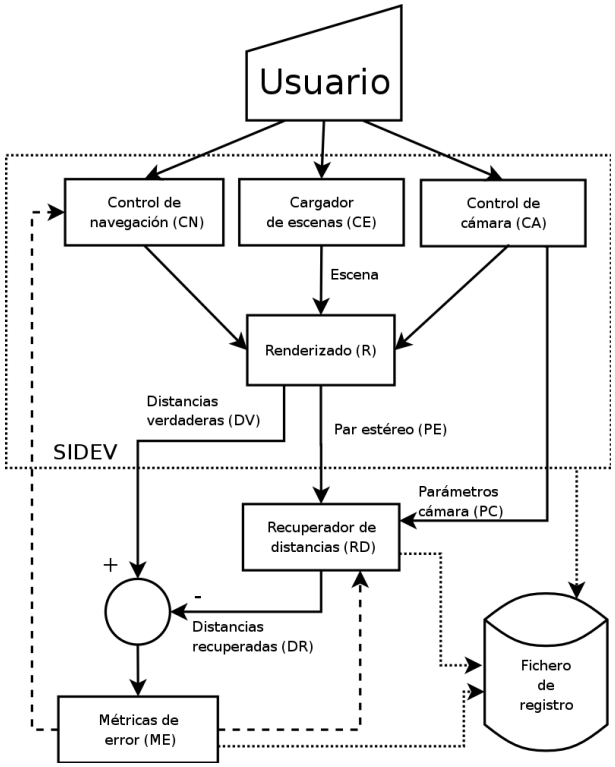


Fig. 1. Esquema del sistema del lazo para la evaluación de algoritmos de reconstrucción estereo.

B. Reconstrucción de distancias estéreo (RD)

La solución del problema de reconstrucción consiste en que conocido un par de imágenes estéreo I e I', a cada píxel p de la imagen I se le asigna un valor de disparidad denotado por d_p, que es inversamente proporcional a la distancia del punto correspondiente de la escena al observador. Este problema se denomina problema de correspondencia y se ha resuelto por medio de diversas aproximaciones[2]. Se ha elegido el algoritmo TRW (Tree Reweighted Message Passing), ya que ha demostrado una precisión mayor que otros algoritmos de paso de mensajes y es paralelizable, haciéndolo más adecuado para su implementación sobre arquitecturas paralelas como las GPU. Se ha realizado su implementación para que pueda leer el flujo contínuo de imágenes provenientes del SIDEV, o de una cámara real respetando las interfaces adecuadas. Esto posibilita la navegación y reconstrucción en tiempo real.

1) El algoritmo TRW: Este algoritmo trata de resolver el problema de correspondencia minimizando una función de energía E, que también puede verse como máxima verosimilitud de la distribución a posteriori de un campo aleatorio de Markov (Markov Random Field)[7]. Esta función de energía E relaciona la energía de los datos E_d y la energía de suavizado E_s mediante la expresión $E = E_d + \lambda E_s$, donde λ determina la importancia relativa de cada término.
La energía de los datos E_d se determina sumando los costos de dato por píxel $c_p(d_p)$, es decir, $E_d = \sum_p c(d_p)$. En este trabajo se toma $c_p(d_p) = \|I(p) - I(p - d_p)\|^2$ donde $I(p)$ e $I(p - d_p)$ son la intensidad del píxel p en la imagen I y la intensidad del píxel $p + d_p$ en la imagen I', teniendo en cuenta que son vectores al tratarse de imágenes en color.

Para determinar la energía de suavizado E_s se asume que los píxeles de la imagen forman una malla bidimensional (grafo) de modo que p está situado en coordenadas $p = (i, j)$ sobre esa malla. Entonces se define una relación de vecindad tal que si $p = (i, j)$ y $q = (s, t)$ entonces $|i - s| + |j - t| = 1$. Denotando por N al conjunto de todas las parejas de píxeles con una relación de vecindad de este tipo, la energía de suavizado se determina como:

$$E_s = \sum_{\{p,q\} \in N} V_{pq}(d_p, d_q) \tag{1}$$

Por otra parte se ha optado por definir $V_{pq}(d_p, d_q)$ como:

$$V_{pq}(d_p, d_q) = \begin{cases} 1 & si \quad d_p \neq d_q \\ 0 & si \quad d_p = d_q \end{cases} \tag{2}$$

El algoritmo trata de encontrar un mínimo de la energía E. Para ello se emplea un método iterativo de paso de mensajes entre los píxeles de la malla bidimensional. Se define $M^t_{p \to q}$ como el mensaje que el píxel p envía a su vecino q en la iteración t, siendo éste vector de tamaño m y m el número de disparidades que se tienen en cuenta. La regla de actualización de los mensajes es la siguiente:

$$M^t_{p \to q}(d_q) = \min_{d_p} \begin{cases} \mu_{pq}(c(d_p) + \sum M^{t-1}_{s \to p}(d_p)) \\ \\ -M^{t-1}_{q \to p}(d_p) + V_{pq}(d_p, d_q) \end{cases} \tag{3}$$

donde los coeficientes μ_{pq} se determinan como sigue: primero, se elige un conjunto de árboles del grafo de vecindad de modo que cada arco esté en al menos un árbol. Luego se elige una distribución de probabilidad ρ sobre el conjunto de árboles, y finalmente, μ_{pq} se fija a ρ_{pq}/ρ_p, es decir, la probabilidad de que un árbol elegido aleatoriamente bajo ρ contenga el arco (p, q) dado que contiene p.
El algoritmo TRW original utilizado en este trabajo no necesariamente converge a la solución óptima. De hecho, no garantiza que el límite inferior decrezca siempre con el tiempo. Este problema se puede reducir utilizando técnicas de amortiguamiento al actualizar los mensajes o usando una versión secuencial del algoritmo, lo que no permitiría la implementación de tiempo real por no estar en consonancia con su implementación sobre hardware con arquitecturas paralelas como la GPU.

2) Implementación sobre GPU del algoritmo TRW: La arquitectura de las GPUs se ajusta bien el paradigma del *stream programming*, basado en la definición de operaciones intensivas en cómputo ejecutadas en paralelo sobre cada uno de los datos, en este caso los píxeles del par de imágenes estéreo. Por este motivo se ha decidido elegir el lenguaje BrookGPU, que sigue este paradigma, para la implementación del algoritmo [8].

Para la implementación se tiene en cuenta la estructura particular de la función $V_{pq}(d_p, d_q)$ elegida. Esta función toma como valor 1 para todos los casos excepto en uno con lo que se puede escribir la regla de actualización como sigue:

$$\begin{gathered} M(d_p) = \mu_{pq}(c(d_p) + \sum M_{s\to p}^{t-1}(d_p)) - M_{q\to p}^{t-1}(d_p) \\ Q(d_q) = \min_{d_q}\{M(d_q) + \lambda\} \\ M_{p\to q}^{t}(d_q) = \min(Q(d_p), M(d_q)) \end{gathered} \tag{4}$$

En la figura 2 se muestra un esquema gráfico de la implementación de la ecuación 4 mediante los elementos del lenguaje BrookGPU.
Con una implementación como la anterior se pierde generalidad en cuanto a la función $V(d_p, d_q)$, sin embargo se gana en eficiencia y se ahorra memoria. Esto es importante, ya que la jerarquía de memoria de la GPU compromete el rendimiento.

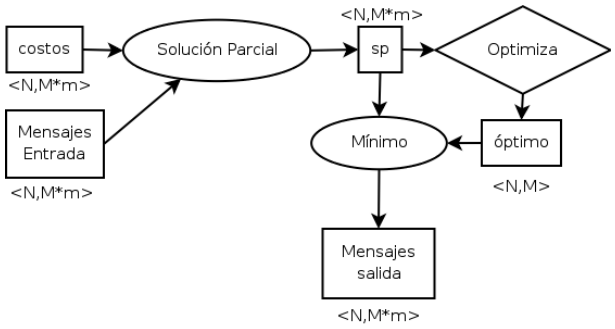


Fig. 2. Esquema de los *kernels* y los *streams* del proceso iterativo.

C. Métricas de calidad (MC)

Puesto que se dispone de mapas de distancias directos generados por el SIDEV, la evaluación se realiza comparando el mapa generado por el algoritmo con el mapa sintético. Se debe tener en cuenta que en los mapas de distancia generados hay varios tipos de píxeles:

- Tipo I, Píxeles de disparidad máxima: Son aquellos que sobrepasan la disparidad máxima, y que corresponden a distancias menores que el mínimo observable.
- Tipo II, Píxeles de disparidad mínima: Son aquellos a los que se les asigna un valor de disparidad menor que la disparidad mínima, y que corresponden a distancias mayores que la distancia máximo observable.
- Tipo III, Píxeles sin correspondencia: Existe un conjunto de píxeles de I que no tienen un píxel asociado en I'. La cantidad de estos píxeles depende de la geometría del sistema óptico y de las distancias presentes en la escena.
- TipoIV, Píxeles correctos.

Al realizar las medidas de calidad es necesario considerar sólo los píxeles de tipo IV puesto que son aquellos que no deberían tener ningún tipo de error asociado [9]:

1) *Error cuadrático medio (R) de las distancias obtenidas $z_c(i,j)$ con respecto a las distancias verdaderas $z_v(i,j)$*

$$R = \left(\frac{1}{N}\sum_{(i,j)}(z_c(i,j) - z_v(i,j))^2\right)^{\frac{1}{2}} \tag{5}$$

donde N corresponde al número de píxeles de tipo IV.
2) *Error cuadrático medio con respecto al mapa de distancias verdaderas cuantizado (RC).*
Los métodos de recuperación estéreo tienen una limitación debida a la naturaleza discreta de las disparidades. Puesto que las distancias son inversamente proporcionales a la disparidad, se obtendrá una mejor resolución en distancias cercanas. Este hecho se debe a la propia formulación del problema y no a un método particular de resolución del problema de correspondencia. Como esta discretización no se da en el problema directo, se propone una métrica adicional que indica en qué medida son erróneas las disparidades calculadas por el algoritmo y cómo afectan esos errores a las medidas de distancia. Para ello se introduce una variante de R que permite medir el error del algoritmo con respecto a un recuperador estéreo perfecto, cuantizando previamente de forma adecuada las distancias $z_v(i,j)$ y aplicando la expresión de la ecuación 5.
3) *Porcentaje de píxeles erróneos (B).*

$$B = \frac{100}{N}\sum_{(i,j)}(|z_c(i,j) - z_v(i,j)| > \delta) \tag{6}$$

donde δ es la tolerancia relativa del error.

III. Experimentos y Resultados

En esta sección se describen las pruebas realizadas para evaluar el rendimiento del sistema de recuperación estéreo. Para ello se tienen en cuenta dos factores, por un lado el tiempo de cómputo y por otro la calidad de los resultados ofrecidos por el algoritmo TRW.

A. Datos de prueba

Para la evaluación del algoritmo se ha seleccionado la escena de interior de la figura 3, que da como resultado el mapa de distancias acotado de la figura 4(a). Se trata de un par de imágenes estéreo de tamaño 100x100 en RGB, que no está sometido a ningún preprocesamiento.

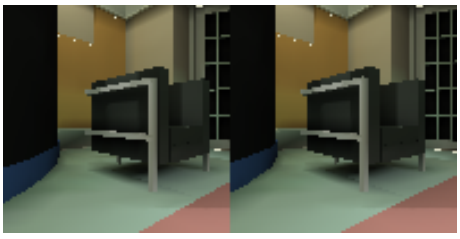
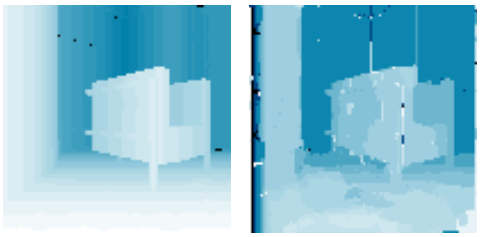


Fig. 3. Ejemplo de un par estéreo generado por el SIDEV utilizado en los experimentos.



(a) Distancias verdaderas

(b) Distancias calculadas con TRW

Fig. 4. Mapas de distancias.

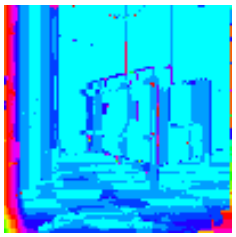


Fig. 5. Mapa de diferencias entre el mapa de disparidades teórico y el generado por el algoritmo TRW.

B. Calidad del algoritmo

Para comprobar la calidad de los resultados obtenidos con el algoritmo TRW se reconstruyen las distancias sobre el par estéreo de la figura 3 obteniéndose el mapa de distancias de la figura 4(b), restringiendo el rango de disparidades a $[-19, 0]$. Para dicha reconstrucción se han calculado las métricas de calidad explicadas en la sección II-C, que se detallan en la tabla I, empleando un conjunto de parámetros óptimo del algoritmo para la escena.

El conjunto óptimo de parámetros a determinar son el valor de λ y el número de iteraciones. El número de iteraciones se estima a partir de las gráficas de la figura 6, obteniéndose un valor estable a partir de las 13 iteraciones, lo que concuerda con los resultados obtenidos en [6]

El valor óptimo de λ se determina a partir de la figura 7, llegando a una zona estable en torno a $\lambda = 4$. Además, hay que hacer notar que el error aumenta para valores altos de λ. Esto se explica por el hecho de que un aumento de λ da lugar a un aumento de la restricción de suavizado sobre los valores de disparidad dando lugar a errores. Para ver en qué zonas de la imagen se cometen errores debido a una restricción de suavizado demasiado fuerte en la figura 5 se ha representado la

Métrica	Valor
R	0.50364
RC	0.49096
B	4.24%
Píxeles de tipo1	0
Píxeles de tipo2	106
Píxeles de tipo3	1700
Píxeles de tipo4	8194

Tabla I
VALOR DE LAS MÉTRICAS DE CALIDAD PARA EL MAPA DE DISTANCIAS DE LA FIGURA 4(B). PARA LA MEDIDA DE B EL VALOR δ SE TOMA COMO EL 10% DE LA DISTANCIA MÁXIMA DE LA ESCENA.

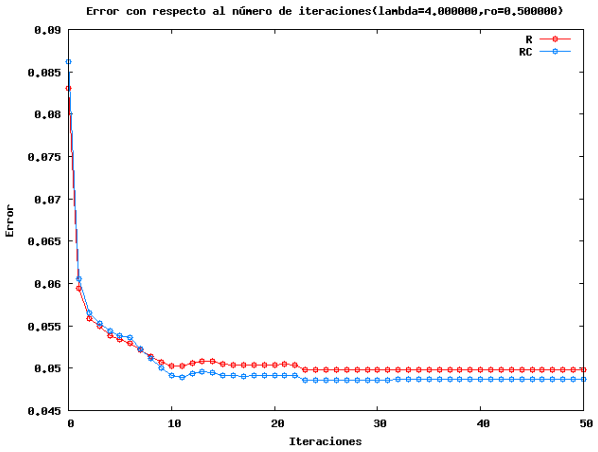


Fig. 6. Errores cometidos según el número de iteraciones para la escena de la figura 3.

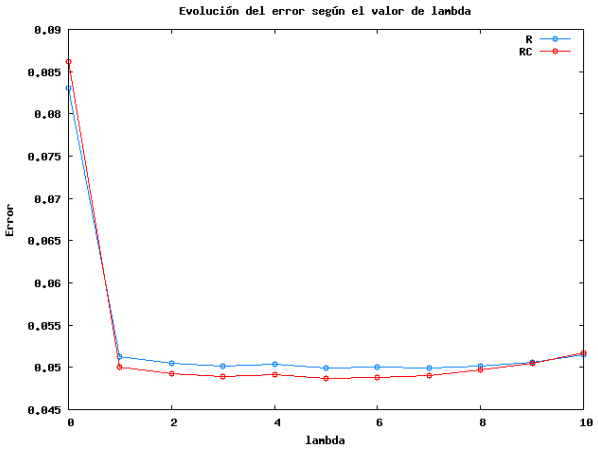


Fig. 7. Errores cometidos al variar λ para la escena de la figura 3 con 13 iteraciones.

imagen de diferencias entre mapa de disparidades obtenido y el mapa de disparidades verdaderas discretizado. En este mapa de diferencias se observa una zona de píxeles de tipo 3 en la parte izquierda. Además de observa que el algoritmo ofrece buenos resultados para zonas con poco cambio en la distancias mientras que en el caso de zonas donde las distancias cambian bruscamente, por ejemplo, en los bordes se producen errores. Para evitar estos errores es necesario relajar la restricción de suavidad sobre las disparidades que se impone mediante la función $V_{pq}(d_p, d_p)$, lo que llevaría a problemas en las zonas uniforme. Esto obliga a establecer un compromiso.
Una vez obtenidos los parámetros óptimos para la escena, se procede a comprobar la sensibilidad al ruido del algoritmo representando en la figura 8 los errores R y RC con respecto a la desviación típica del ruido gausiano que se introduce en las imágenes. Se puede observar que el algoritmo es sensible al ruido a partir de una desviación típica de 0.7. A partir de este valor el error aumenta a mayor desviación típica del ruido.

C. Tiempo de cómputo

Se ha medido el tiempo de cómputo para el flujo de imágenes generado por el SIDEV. Este flujo está regulado por un protocolo de parada y espera, que permite ajustar el ritmo

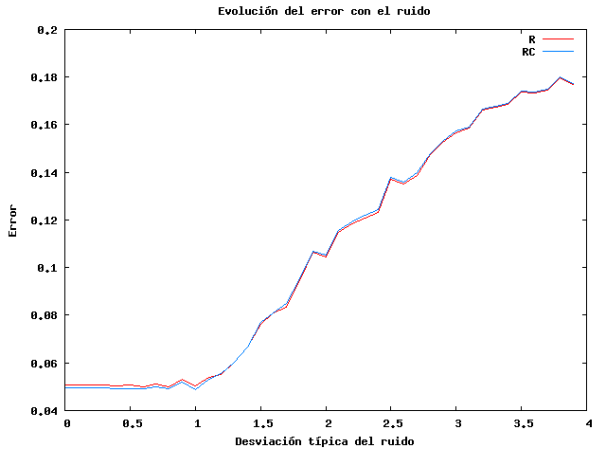


Fig. 8. Errores cometidos según la desviación típica del ruido gausiano para la escena de la figura 3.

Tamaño	Tiempo(s)	Frames/s
32x32	0.048	20
64 x 64	0.053	18
75 x 75	0.139	7
100 x 100	0.144	6

Tabla II
TIEMPOS DE CÓMPUTO SEGÚN EL TAMAÑO DE LAS IMÁGENES AL PROCESAR EL FLUJO PROPORCIONADO POR EL SIDEV CON 20 NIVELES DE PROFUNDIDAD.

del mismo a la velocidad de procesamiento del algoritmo al que se acopla. Las pruebas se han realizado sobre una GPU de NVIDIA 6600 GTX, la cual es de gama media, en un PC con un procesador AMD Athlon(tm) 64 3500+ y 2GB de memoria RAM. En la tabla II se presentan los tiempos por *frame* alcanzados según el tamaño de las imágenes, limitando el número de niveles de profundidad a 20. Los valores mostrados en la tabla indican el rendimiento esperable para cada tamaño de imagen, alcanzándose cifras adecuadas para su uso con cámaras reales.

IV. Conclusiones

Se ha llevado a cabo la implementación del algoritmo de recuperación estéreo TRW sobre GPU, quedando demostrada la potencia de este tipo de hardware para este tipo de aplicaciones. Además se abre la posibilidad de ejecución en tiempo real de un algoritmo que ofrece un elevada precisión en ausencia de ruido. Sin embargo, aplicado a imágenes con ruido la precisión del mismo disminuye. Éste es un factor que hay que tener en cuenta a la hora de emplearlo con imágenes procedentes de cámaras reales, que probablemente deban ser sometidas a algún tipo de preprocesado con el fin de eliminar las altas frecuencias que introduce el ruido de este tipo. Esta eliminación de potencia de alta frecuencia ayudaría a mejorar las tasas de error en las zonas de borde.

Agradecimientos

Este trabajo está subvencionado parcialmente por la Fundación Loro Parque y el “Programa Nacional de Diseño y Producción Industrial (Proyecto DPI 2006-07906) del Ministerio de Educación y Ciencia de España y el Fondo Europeo de DEsarrollo Regional (FEDER)”.

Referencias

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[2] M. Wainwright, T. Jaakkola, and A. Willsky, “Map estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches,” vol. 51, no. 11, pp. 3697–3717, November 2005.

[3] V. Kolmogorov and C. Rother, “Comparison of energy minimization algorithms for highly connected graphs.” in *ECCV (2)*, 2006, pp. 1–15.

[4] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother, “A comparative study of energy minimization methods for markov random fields.” in *ECCV (2)*, 2006, pp. 16–29.

[5] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, “A survey of general-purpose computation on graphics hardware,” *Computer Graphics Forum*, vol. 26, no. 1, pp. 80–113, 2007. [Online]. Available: http://www.blackwell-synergy.com/doi/pdf/10.1111/j.1467-8659.2007.01012.x

[6] J. G. Marichal-Hernández, F. P. Nava, F. Rosa, R. Restrepo, and J. M. Rodríguez-Ramos, “An integrated system for virtual scene rendering, stereo reconstruction and accuracy estimation.” in *GMAI*, 2006, pp. 121–126.

[7] V. Kolmogorov and M. Wainwright, “On the optimality of tree-reweighted max-product message-passing,” in *Proc. e on Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, 2005.

[8] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Mike, and H. Pat, “Brook for gpus: Stream computing on graphics hardware,” 2004. [Online]. Available: citeseer.ist.psu.edu/buck04brook.html

[9] D. Scharstein, R. Szeliski, and R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” 2001. [Online]. Available: citeseer.ist.psu.edu/scharstein01taxonomy.html