ARQUITECTURA BASADA EN FPGA PARA LA RECUPERACIÓN ESTÉREO EN TIEMPO REAL PARA UNA CÁMARA INTELIGENTE.

Por:

Víctor Manuel García y García

Tesis sometida como requisito parcial para obtener el grado de Maestro en Ciencias en la especialidad de Ciencias Computacionales en el Instituto Nacional de Astrofísica, Óptica y Electrónica.

Supervisada por:

Dr. Miguel O. Arias Estrada.

Resumen

La visión estéreo permite calcular o construir una estructura tridimensional de una escena a partir de dos o más imágenes tomadas desde distintos puntos de vista. La idea básica de los algoritmos estéreo es encontrar el punto de una escena capturada por un sensor de imagen con su respectivo punto proyectado en otro sensor de imagen. El procesamiento de búsqueda requiere de un alto número de operaciones. La ejecución de las operaciones, así como el acceso a los datos, representan limitantes para aplicaciones en tiempo real.

En este trabajo, se propone un sistema de visión estéreo para la recuperación tridimensional en tiempo real, empotrado en un dispositivo FPGA. Al integrar un par de sensores estéreo con el dispositivo FPGA que contiene la arquitectura hardware se obtiene una cámara inteligente 3D. La arquitectura hardware se diseñó analizando los datos independientes con respecto al tiempo, los resultados de rendimiento, muestran que la arquitectura propuesta permite una velocidad de procesamiento de 30 cuadros por segundo usando imágenes de tamaño 640x480 píxeles. Mientras que los resultados de implementación muestran que el uso del FPGA es del 60% (6,580 flip-flops). La comparación entre arquitecturas hardware no es trivial, sin embargo, se puede utilizar el número de píxeles procesados por segundo como métrica de comparación. Usando la métrica anterior, se determina que la arquitectura propuesta tiene mejor desempeño que las encontradas en la literatura. Una de las aportaciones de la arquitectura hardware es la configuración variable en tamaño de imágenes, tamaño de ventana de búsqueda y tamaño de rango de disparidad. Cualquier configuración alcanza la velocidad de procesamiento en tiempo real, sin embargo, el uso de recursos en el FPGA aumenta.

Summary

Stereo vision allows to calculate a tridimentional structure of a scene from two or more captured images taken from diferents points of view. The basic idea of stereo algorithms is to find the point of one scene captured with a image sensor with its respective point projected in another image sensor.The search proccesing requieres a high number of operations. A real time application is limited by the execution of those operations and the data access.

In this work is proposed a real-time 3D recovery stereo vision system, embebed on a FPGA. The integration of a FPGA based hardware arquitecture with a pair of stereo images sensors results into a 3D smart camera. The hardware arquitecture design was based on an independient-time data analisys. The performance results shows that the propose arquitecture can process 30 frame per second with 640x480 pixels images. The implementation results shows that a 60% FPGA resource were used (6,580 flip-flops). The hardware arquitecture comparative is not easy to stablish, however, we can use the processed pixels per second number as a comparation metric. Using this metric, it can be stablished that the arquitecture in this work has a better performance than the found in the literature. One of the hardware arquitecture contribution is the variable performance configuration based on images sizes, search window size and disparity range size. Any configuration achieves a real time performance, but the FPGA usage increases.

Agradecimientos.

Agradezco al CONACYT por el apoyo económico otorgado para realizar mis estudios de maestría. Al Instituto Nacional de Astrofísica, Óptica y Electrónica por el apoyo en instalaciones.

Deseo agradecer al Dr. Miguel O. Arias Estrada por su apoyo, consejos y conocimientos necesarios para la realización de este trabajo. Agradezco también a Gerardo Sosa por su gran ayuda y paciencia que me brindó. Hago extenso el agradecimiento al comité encargado de la revisión de mi tesis por el tiempo dedicado y por las observaciones recibidas.

Por último doy gracias a mis compañeros de generación en especial a Eliezer, Felix y Olmo, ya que con ellos aprendí muchas cosas más que los conocimientos científicos.

A mis padres:

Profa. Margarita García Bello.
Prof. Luis Roberto García Hernández.

A mis hermanos:

Ing. Luis Roberto García y García.
Dr. Víctor Hugo García y García.

A mi esposa:

Dr. Angélica Ponce Gómez.

A la familia Ponce Gómez:

Lic. Fernando Ponce Ceballos.
Sra. María Guadalupe Gómez Pérez.
Ing. Eder Ponce Gómez.

Tabla de contenido

Índice de figuras.

Índice de tablas.

Capítulo 1

Introducción.

1.1 Descripción del problema.

Los tipos de sistemas compuestos por uno o más sensores de imágenes y un dispositivo de procesamiento empotrado son comúnmente llamados “Cámaras inteligentes” [26]. El objetivo de una cámara inteligente es el procesamiento integrado a nivel chip, esto es, que la carga computacional del procesamiento de bajo nivel se efectúe en el dispositivo empotrado. Este dispositivo debe procesar imágenes automáticamente con el fin de obtener información relevante del escenario capturado. Las cámaras inteligentes deben realizar sus operaciones con restricciones de tiempo, de recursos de cómputo, de memoria y de potencia. El bajo costo y el bajo consumo de potencia de los dispositivos digitales, los convierten en plataformas interesantes para desarrollar sistemas empotrados. Una aplicación de estos sistemas es la visión por computadora, en específico, la visión estéreo.

La palabra estéreo es comúnmente relacionada con el sonido, pues la mayoría de los reproductores de música describen la calidad del sonido que reproducen como sonido estéreo. Es un error popular relacionar la palabra estéreo con un par de objetos cualesquiera que sean. Estéreo, según la Real Academia Española, significa ‘Sólido’, y hace referencia al estado de la materia sólida es decir, cuerpos definidos en tres dimensiones.

La visión estéreo, por consiguiente, se refiere a la adquisición de imágenes para ser interpretadas en tres dimensiones. La visión estéreo está presente en los seres humanos en el sentido de la vista. Los seres humanos interpretan una escena vista por sus ojos, de manera tridimensional, gracias

al procesamiento que se realiza en la corteza visual del cerebro [11]. El sistema visual de los seres humanos se basa en un par de sensores biológicos de imágenes (los ojos), sin embargo, la visión estéreo no está limitada a sólo 2 sensores.

La visión estéreo permite calcular o construir una estructura tridimensional de una escena a partir de dos o más imágenes tomadas desde distintos puntos de vista. La idea básica de los sistemas de visión estéreo se basa en el hecho de que un punto en el espacio tridimensional se proyecta a un único par de localidades en dos sensores. Por lo tanto, idealmente es posible determinar las localidades que corresponden a un mismo punto en el espacio. De lo anterior se establece que es posible encontrar la estructura tridimensional. En la práctica, existen limitaciones por las que no siempre es posible determinar cuales son las proyecciones únicas de un punto de la escena capturada. Una de estas limitaciones ocurre cuando un sensor está parcialmente ocluido, esto es, que la escena que se captura, está parcialmente obstruida por otro objeto, por lo que no será posible determinar todas las proyecciones únicas. Sensores con diferentes características físicas generan imágenes con diferencia en los valores de intensidad de los píxeles, lo cual genera ambigüedad para encontrar las proyecciones únicas. Otra limitante son las distorsiones de proyección que ocurren al capturar las escenas, entre otras.

En este trabajo se propone un sistema de visión estéreo para la recuperación tridimensional en un dispositivo digital del tipo arreglos de compuertas lógicas programables en el campo, es decir dispositivos FPGAs. Al integrar un par de sensores estéreo con el dispositivo FPGA que contiene la arquitectura hardware que realiza el procesamiento para recuperar la estructura tridimensional, se obtiene una cámara inteligente 3D.

Anteriormente, se menciona que la idea básica para los algoritmos estéreo es encontrar el punto de una escena capturada por un sensor de imagen con su respectivo punto proyectado en otro sensor de imagen. Al conocer las posiciones distintas del mismo punto se puede establecer la diferencia en distancia de esas posiciones. A esta diferencia, se le conoce como disparidad. Los algoritmos que calculan las disparidades de dos o más imágenes se les conocen como algoritmos de correspondencia estéreo.

Sin embargo, para poder calcular la correspondencia estéreo, es necesario resolver otros problemas, entre ellos, eliminar las distorsiones presentes en las imágenes capturadas debido a la óptica de los sistemas de captura. También es necesario cumplir con restricciones geométricas para poder realizar la búsqueda de correspondencia en tiempos aceptables para diversas aplicaciones.

La investigación de sistemas de visión estéreo ha tomado un gran auge durante la última década, debido principalmente a las necesidades tecnológicas de diversas aplicaciones, como son: segmentación automática, seguimiento, análisis de estructuras u objetos, vigilancia [7,6] realidad virtual, sistemas médicos [15], entre otros.

Existen diferentes sistemas de visión estéreo que se pueden encontrar en la literatura, sin embargo, la mayoría son implementaciones en sistemas de cómputo de propósito general, estas implementaciones son limitadas por el procesamiento secuencial de los sistemas de cómputo tradicional, entre otras causas como el manejo de memoria realizado por el sistema operativo. También existen implementaciones en sistemas de cómputo de propósito específico que aceleran el algoritmo de correspondencia, sin embargo, la mayoría no realizan las transformaciones geométricas en las imágenes cap-

turadas, dichas transformaciones son necesarias para obtener resultados que describan de mejor forma la estructura tridimensional.

1.2 Objetivos.

Objetivo general

Explorar algoritmos para la recuperación 3D en tiempo real y proponer una alternativa arquitectural basada en FPGA para una cámara inteligente.

Objetivos específicos.

- Revisión de algoritmos que realizan la corrección geométrica de un par de imágenes estéreo y adaptarlo a una arquitectura hardware.
- Revisión de algoritmos para resolver el problema de correspondencias usando correlación estéreo.
- Analizar diversos algoritmos para el cálculo de medidas de similitud y adaptar los seleccionados a una arquitectura hardware reconfigurable.
- Proponer un modelo hardware que resuelva la gestión de accesos a memoria de manera eficiente.
- Diseñar e implementar un modelo hardware que sincronice los modelos anteriores

1.3 Organización de la tesis.

En este trabajo, se realiza una investigación sobre las transformaciones geométricas necesarias para el algoritmo de correspondencia. También se analizan los distintos algoritmos de correspondencia estéreo con el fin de proponer una arquitectura hardware basada en FPGA para la recuperación

tridimensional para una cámara inteligente. Se propone el diseño de la arquitectura que cumple con diversas restricciones, entre las más importantes está que la arquitectura transforma las imágenes recibidas del par de los sensores estéreo con el fin de que cumplan la restricción epipolar, Dicha restricción es necesario cumplirla para poder acelerar la búsqueda de correspondencias al reducir el rango de búsqueda a la línea epipolar. La arquitectura propuesta es capaz de entregar mapas de disparidad con una velocidad de video; es decir 30, cuadros por segundo, cumpliendo con la restricción de tiempo real.

La arquitectura hardware desarrollada en este trabajo puede ser complementada con la integración de módulos de post-procesamiento, en específico, se propone como extensión a la arquitectura hardware, la integración de módulos de interpolación que permitirán obtener mapas de disparidad de mejor calidad. Otra extensión propuesta se basa en el uso de imágenes a color para su procesamiento estéreo. Se ha demostrado estadísticamente [8] que el uso de imágenes a color mejora la calidad de los mapas de disparidad denso.

Por otro lado, la arquitectura propuesta en este trabajo cuenta con características importantes, principalmente la escalabilidad de la misma. A pesar que la arquitectura estéreo propuesta trabaja con parámetros fijos de búsqueda para el algoritmo de correspondencia, la arquitectura tiene la capacidad de rediseñarse de una manera relativamente fácil con el fin de modificar los parámetros de búsqueda, manteniendo la velocidad de procesamiento. Mas adelante se mostrará como resultado de este trabajo de investigación, la implementación de una arquitectura hardware basada en FPGA que permite la recuperación 3D en tiempo real dado un par de imágenes estéreo sin rectificar. La arquitectura hardware se basa en elementos procesador (PE). El número de PEs puede ser variado para cumplir diversas restricciones de la aplicación final.

El documento está dividido por capítulos de la siguiente manera. El capítulo 2 presenta los fundamentos teóricos en donde se basa la investigación de este trabajo. El capítulo 3 describe el diseño de la arquitectura estéreo que permite generar mapas de disparidad a partir de un par de imágenes estéreo no rectificadas. El capítulo 4 muestra los resultados de la implementación, así como los resultados de rendimiento y los resultados de prueba que validan el funcionamiento de la arquitectura estéreo. El documento termina en el capítulo 5 con las conclusiones obtenidas por el desarrollo de esta investigación, así como trabajos a futuro sugeridos para el crecimiento de la arquitectura estéreo.

Capítulo 2

Fundamentos teóricos

2.1 Introducción.

La visión estéreo permite calcular o construir una estructura tridimensional de una escena a partir de dos o más imágenes tomadas desde distintos puntos de vista. La idea básica de los sistemas de visión estéreo se basa en el hecho de que un punto en el espacio tridimensional se proyecta a un único par de localidades en dos cámaras. Por lo tanto, idealmente es posible determinar las localidades que corresponden a un mismo punto en el espacio. Al conocer las localidades donde un mismo punto se proyecta en 2 o más imágenes es posible encontrar la estructura tridimensional de la escena. En la práctica, existen limitaciones por las que no siempre es posible determinar cuales son las proyecciones únicas de un punto de la escena capturada. Una de estas limitaciones ocurre cuando un sensor está parcialmente ocluido, esto es, que la escena que se captura, esta parcialmente obstruida por otro objeto, por lo que no será posible determinar todas las proyecciones únicas. Por otro lado, existen ambigüedades que no siempre permiten encontrar las correspondencias estéreo, dichas ambigüedades son generadas por diversos factores como diferencia en texturas de los objetos capturados así como imágenes no ecualizada, es decir con imágenes de intensidades normalizadas.

Sin embargo, cuando se es posible conocer las proyecciones de todos los puntos de la escena en ambas cámaras, se puede generar un mapa basado en las diferencias de las proyecciones a lo que se le conoce como **mapa de disparidad**. Si se determinan todas las disparidades de la escena capturada, entonces el mapa de disparidad es *denso*. El mapa de disparidad se

puede calcular a través de distintos métodos llamados *algoritmos de correspondencia.* Independientemente de la técnica o algoritmo usado para generar el mapa de disparidad, la reconstrucción tridimensional se basa en este mapa junto con el conocimiento de la geometría del sistema estéreo.

La fiabilidad de los resultados generados por los algoritmos de correspondencia depende de distintos factores, en particular, es necesario que las imágenes del sistema estéreo estén rectificadas, esto es, que cumplan con la restricción epipolar logrando realizar la búsqueda de correspondencias solamente sobre la línea definida por los epipolos (línea epipolar). A lo largo de este capítulo, se revisan los fundamentos teóricos en los que se basa la visión estéreo, como lo es la adquisición y formación de imágenes, geometría estéreo y epipolar, calibración y el problema de correspondencia. Una vez presentadas estas bases, se repasan las arquitecturas hardware, de donde se sugiere que una implementación en hardware basada en FPGA podrá acelerar el algoritmo estéreo. En la sección 2.6 se realiza una discusión acerca de las limitaciones y perspectivas de este trabajo.

2.2 Adquisición y formación de imágenes.

Las imágenes estéreo se pueden generar de diversas maneras en donde parámetros geométricos y ópticos definen al sistema que las captura. Debido a las distintas configuraciones de los sistemas de captura estéreo, se han realizado trabajos [24] que analizan la geometría y la posición de los sensores para optimizar la recuperación estéreo. También se puede controlar la variación en la iluminación de la escena, a este método se le conoce como *fotometría estéreo* [29]. Debido a la gran cantidad de factores a controlar, en la práctica estos tipos de sistemas son limitados en sus aplicaciones.

Otro método para la recuperación tridimensional, se basa en la variación del foco de la lente de la cámara. Si se adquieren varias imágenes del mismo objeto, cada imagen capturada con distintos enfoques, entonces se puede recuperar la información tridimensional del objeto. A dicho método se le conoce como "Profundidad a partir del enfoque" *(depth from focus/defocus)*. Se puede revisar [17] para mayor información de este método.

También se puede recuperar imágenes estéreo simulando un sistema estéreo usando una sola cámara. Es necesario capturar una primera imagen y después trasladar la cámara para capturar la segunda. El inconveniente de este método es que el objeto debe estar estático. Otra limitación surge ya que es necesario un montaje robotizado que realice la traslación de la cámara con el fin de realizar la calibración; sin embargo, el proceso de calibración debe entregar resultados milimétricos, los cuales no siempre se logran sin un previo procesamiento que corrija las distorsiones en las imágenes.

En este trabajo se estudiarán los sistemas estéreo basados en dos cámaras. En la siguiente sección se describe la importancia de la geometría en el procesamiento de los sistemas estéreo, así como las correcciones a las diversas distorsiones sufridas durante el proceso de captura.

2.3 Visión estéreo.

2.3.1 Geometría monocular.

Existen distintos dispositivos de captura de imágenes, desde los ojos en los animales hasta cámaras de video y radio telescopios. Muchos de estos dispositivos están equipados con lentes. Tómese como primer ejemplo la cámara obscura que fue inventada en el siglo XVI. Esta cámara no tenía len-

tes, en su lugar, usaba un agujero del tamaño de un alfiler o *pinhole*, el cual enfocaba los rayos de la luz hacia un plano translucido. Esta proyección de los rayos de luz, representa la imagen del escenario capturado por el pinhole. En la actualidad, las cámaras digitales son en esencia, una cámara obscura capaz de grabar la cantidad de luz que llega a una pequeña área de su plano anterior.

Supongamos una caja con un pequeño agujero en el centro de una de sus caras; y con un plano translucido en la cara contraria al agujero. Si pudiéramos colocar esta caja en un cuarto con poca luz, en donde existiera una fuente de luz, digamos, una vela; al colocar la caja con el agujero en dirección a la vela, se observaría una imagen invertida de la vela proyectada en el plano translucido (Figura 2.1). Esta imagen es formada por los rayos de la luz emitidos por la escena, los cuales pasan por el agujero de la caja y se proyectan en el plato translucido.

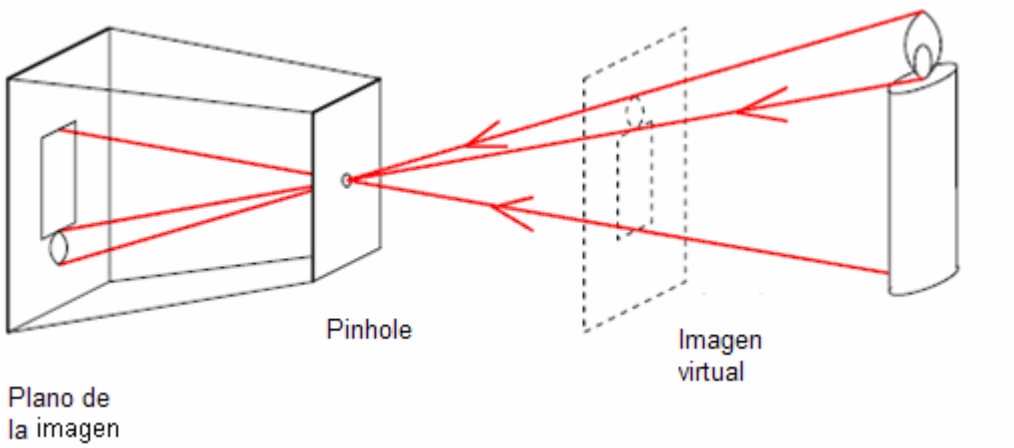

Figura 2.1 Modelo de la cámara pinhole.

El proceso anterior que menciona a una caja con un agujero, se le conoce como cámara *pinhole*. En realidad, el agujero tiene un tamaño finito, por lo que cada punto en el plano de la imagen agrupa luz desde un cono de rayos a cierto ángulo. Por otro lado, las cámaras reales están equipadas con

lentes, lo que complica el modelado del proceso de captura. En [23], se hace una discusión de las similitudes del modelo de pinhole con respecto al modelo real; donde en resumen, se concluye que el modelo de proyección de perspectiva pinhole es matemáticamente conveniente y a pesar de su simplicidad provee una aproximación aceptable al proceso de captura de imágenes.

Existen dos principales razones que justifican el uso de lentes en cámaras reales. La primera razón es para concentrar mejor la luz, dado que un rayo de luz debería alcanzar un solo punto en el plano de la imagen bajo un modelo ideal pinhole. Como ya fue discutido, el tamaño del agujero es finito, esto ocasiona que mientras más grande sea el agujero, más ancho será el cono de rayos por lo que la imagen tendrá más brillo, sin embargo, esto produce que las imágenes se vuelvan borrosas. Por otro lado, el reducir el tamaño del agujero produce imágenes que representan sus bordes de manera más fina pero reduce la cantidad de luz que llega al plano de la imagen, lo que puede producir efectos de difracción. La segunda razón para usar lentes, es el mantener la imagen enfocada finamente mientras se concentra la luz de un área grande.

Como se explicó anteriormente, el modelo de la cámara de agujero de alfiler realiza una transformación de perspectiva del espacio tridimensional al plano de la imagen *R* a través de su centro óptico *C*. Se define el espacio tridimensional a través del sistema de coordenadas (C, x, y, z) y al plano R con las coordenadas (c, u, v). El eje *z* es perpendicular al plano *R* y se llama eje óptico de la cámara (Figura 2.2).

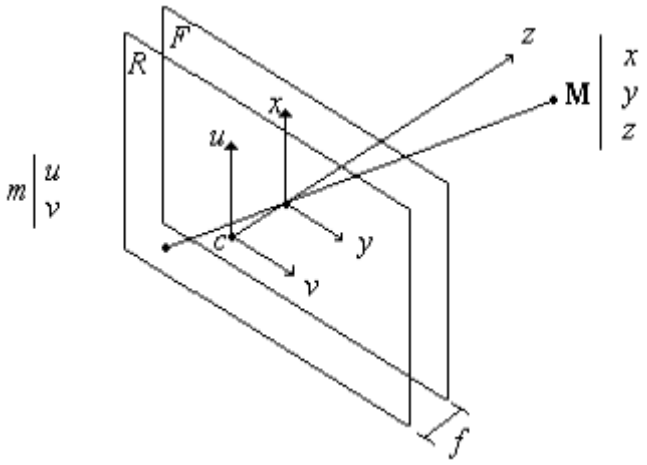

Figura 2.2 Sistema de coordenadas. Proyección 3D del punto M.

De la figura 2.2 se obtiene que la relación entre las coordenadas de la imagen y las coordenadas tridimensionales del espacio pueden escribirse como:

$$-\frac{f}{z}=\frac{u}{x}=\frac{v}{y} \tag{2.1}$$

Al reescribir la ecuación 2.1 de forma lineal, entonces se puede establecer como sigue:

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.2}$$

Donde $u=\frac{U}{S}$, $v=\frac{V}{S}$ y $S=z$.

Se observa de lo anterior que sólo cuando *z* es distinto de cero, las coordenadas *u, v* están definidas.

2.3.2 Modelo de Cámaras geométricas.

Revisando la geometría de la adquisición y formación de las imágenes, se observa que un punto de la escena capturada se proyecta a un punto en el plano de la imagen. Sin embargo, una línea capturada en una perspectiva paralela al agujero de la cámara (en adelante será llamado centro óptico) se proyectará como un punto en el plano de la imagen, de la misma manera se comporta un plano observado bajo la misma perspectiva proyectándose como una línea en el plano de la imagen. Sin embargo, generalmente, una pequeña perturbación en la perspectiva restaura la proyección correcta.

Es importante tener en cuenta estas consideraciones geométricas cuando se realiza procesamiento de imágenes, en específico, en visión computacional. La importancia de las consideraciones geométricas discutidas anteriormente reside en el hecho de poder determinar restricciones cualitativas de una escena en lugar de suposiciones hechas por la relativa configuración de la cámara que hizo la observación. Es igual de importante establecer restricciones cuantitativas entre las medidas realizadas en la imagen y la orientación y posición de las figuras geométricas tomadas en coordenadas de un sistema exterior.

A continuación se describen los parámetros que definen a una cámara geométrica, dichos parámetros se dividen en internos (son aquellos que definen la geometría de la cámara) y externos (son aquellos que definen la posición y orientación de la cámara con respecto al mundo real).

Parámetros geométricos de una cámara.

En la práctica, el sistema de coordenadas de la cámara y del mundo real, se relacionan a través de parámetros físicos tales como la longitud focal

de la lente de la cámara, el tamaño de los píxeles capturados por el sensor CCD o CMOS, la posición del centro óptico y la posición y orientación de la cámara.

En las siguientes secciones se identifican estos parámetros, donde se define como parámetros internos aquellos que establecen la geometría de la cámara y como parámetros externos los que definen la posición y orientación de la cámara con respecto al mundo real.

- Parámetros internos
    - Longitud focal de la lente de la cámara
    - Kappa. Coeficiente de distorsión radial generada en las imágenes capturadas. Dicha distorsión es generada por la estructura física de la lente de la cámara.
    - Tamaño del píxel. Los sensores que capturan los rayos de luz provenientes de la escena están definidos por el tamaño del píxel que representa dicha luz.
    - Coordenadas del centro óptico de la lente de la cámara.
- Parámetros externos.
    - Parámetros de translación. Definen la transformación en movimiento de translación que sufre la imagen capturada con respecto a las coordenadas del mundo real.
    - Parámetros de rotación. Definen la transformación del movimiento de rotación que sufre la imagen capturada con respecto a las coordenadas del mundo real.

2.3.3 Distorsiones en las cámaras.

Existen distintos tipos de distorsiones que sufren las imágenes generadas por una cámara. Estas distorsiones son generadas principalmente por las pequeñas imperfecciones en las superficies de las lentes que capturan los rayos de luz provenientes del escenario. Sin embargo, existen otros factores que producen distorsiones, como lo son la posición del sensor de la cámara, el movimiento interno o vibraciones en la cámara y por último por la temperatura.

Durante el proceso de la captura del escenario por una cámara para generar una imagen, existe una característica propia de estos sistemas de captura, llamada proyección de perspectiva. Esta característica provoca que los ángulos y las distancias reales del escenario no se conserven en la imagen capturada. La proyección de perspectiva también puede ser clasificada como una distorsión. La distorsión causada por la proyección de perspectiva es una distorsión lineal ya que puede ser expresada como una matriz lineal. Por otro lado, las lentes producen otro tipo de distorsión, ésta, es causada por las características físicas de la lente. Dependiendo de la curvatura de la lente, los rayos de luz de la escena son capturados de forma no lineal. Esto es, los rayos que entran por el zona central de la lente se mantienen uniforme, sin embargo mientras más se alejen los rayos de luz del centro de la lente, éstos son capturados de forma distorsionada. A esta distorsión se le conoce como distorsión radial, donde el desplazamiento es una función no lineal que se define a través del centro óptico y la simetría circular de la lente.

En la práctica se observan dos efectos causados por la distorsión radial. El primer efecto es cuando los puntos son llevados hacia el centro de la imagen; esta distorsión es llamada de barril. El segundo efecto es cuando los puntos de la imagen son llevados desde el centro hacia las orillas; a esta dis-

torsión se le llama de almohadilla. La figura 2.3 muestra estos efectos sobre las imágenes.

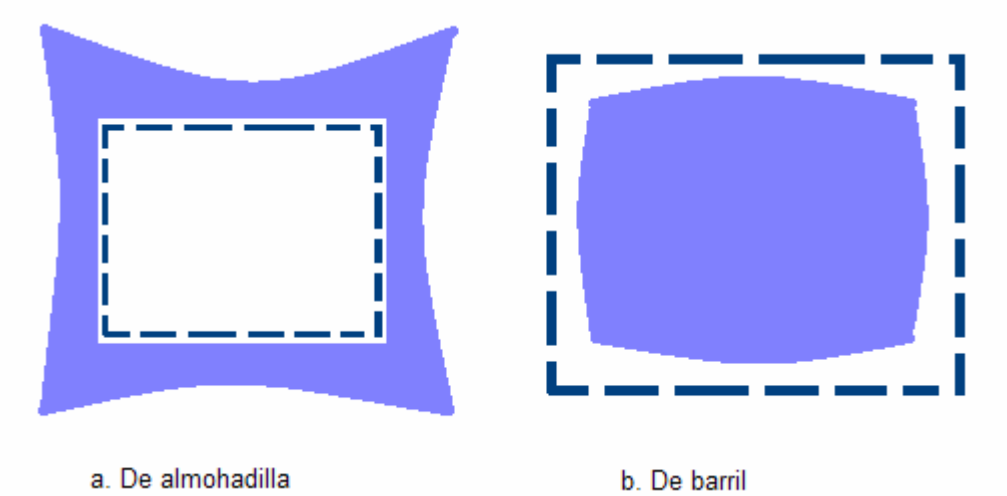

Figura 2.3. Distorsiones radiales.

Distorsión radial.

Diversos algoritmos de la visión computacional como los algoritmos de fotometría, seguimiento, detección de características [22], entre otros, generan mejores resultados al trabajar con imágenes que representen de forma fidedigna la escena capturada. Dicho escenario se captura a través de una lente que permite pasar rayos de luz de la escena. Sin embargo, por las características físicas de las lentes, esta imagen sufre de varias distorsiones, entre las cuales se encuentra la distorsión radial. La figura 2.4 muestra un ejemplo de una imagen distorsionada (radialmente) y su corrección.

Figura 2.4a. Imagen con distorsión radial.

Figura 2.4b. Imagen corregida.

Una solución para evitar la distorsión radial es ocupando lentes de mayor calidad que mantengan la misma resolución tanto en la zona central (fóvea) como en las orillas de la imagen, sin embargo, esta solución agrega un costo significativo al equipo de captura.

Otra solución es la algorítmica, trabajar con una imagen digital y aplicarle un algoritmo que elimine dicha distorsión radial.

Para implementar el algoritmo que elimina la distorsión radial (y en realidad, la mayoría de los algoritmos de visión computacional), se puede realizar en un sistema de cómputo de propósito general, pero esto representa limitación en tiempo y memoria debido a las características del sistema de cómputo. Una discusión más detallada acerca de las limitantes de la implementación de los algoritmos de visión computacional en sistemas de cómputo de propósito general se presenta en el capítulo 3.

Otra opción para la implementación del algoritmo que elimina distorsiones en imágenes, es usando una arquitectura hardware basada en FPGA [10]. La implementación de cualquier algoritmo en arquitecturas específicas (DSP, sistemas de cómputo de propósito general) está limitada por la resolución de datos binarios. Por otro lado, la descripción de hardware en un FPGA permite diseñar y trabajar con datos con la resolución adecuada para el algoritmo, por lo tanto la implementación de algoritmos en arquitecturas específicas requieren por lo general, una simplificación.

Corrección radial.

Para realizar la corrección radial se siguen los siguientes pasos:

1. Calibración geométrica de la cámara.
   En este proceso, el objetivo es encontrar los parámetros geométricos de la cámara. La generación de estos parámetros se debe a que las ecuaciones que modelan la corrección radial son dependientes al factor de distorsión, dicho factor es uno de los parámetros internos de la

cámara. El proceso de la calibración es discutido con más detalle en la sección 2.3.6

2. Modelar la corrección radial.
   Este modelo matemático se representa mediante las siguientes ecuaciones:

$$\begin{aligned} x &= x_d\left(1+\alpha_1 r^2+\alpha_2 r^4+\alpha_3 r^6\right), \\ y &= y_d\left(1+\alpha_1 r^2+\alpha_2 r^4+\alpha_3 r^6\right), \end{aligned} \qquad (2.3)$$

   donde (x_d, y_d) son las coordenadas del punto distorsionado, $r^2 = x_d^2 + y_d^2$ es la distancia del punto distorsionado hacia el punto central y α_1, α_2, α_3, son los coeficientes de distorsión de la cámara que se obtienen a través de la calibración de la cámara.

La corrección radial se realiza sobre las coordenadas de la imagen y no sobre los valores de intensidad de sus píxeles. Observando las ecuaciones que modelan la corrección radial, se sigue que este proceso, así como el anterior (calibración de la cámara), se puede realizar fuera de línea, Por lo tanto, la corrección radial se puede implementar mediante un mapeo de coordenadas. Dada una coordenada de la imagen real, realizar un mapeo hacia la coordenada corregida.

Por lo tanto, la idea es generar, mediante el cálculo de las ecuaciones que modelan la corrección radial, las tablas que mantengan el registro de las coordenadas reales con sus correspondientes coordenadas corregidas. Esta idea acelera el algoritmo de corrección radial, pues el cálculo de las tablas de mapeo sólo realiza una sola vez mientras que el mapeo de coordenadas se realiza cada vez que se capture una nueva imagen. La figura 2.5 muestra el procedimiento de forma gráfica.

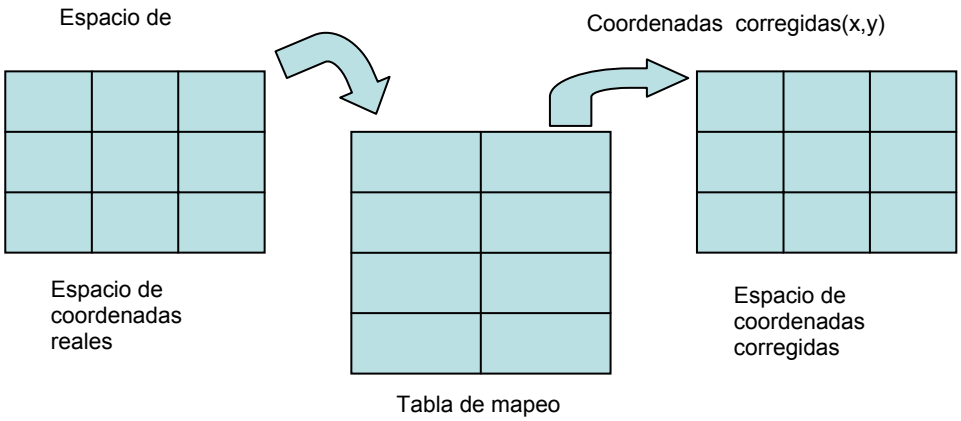

Figura 2.5 Corrección radial mediante tablas.

2.3.4 Geometría estéreo.

Como se describió en la sección anterior, los parámetros geométricos de la cámara son una representación de las características del sistema de captura. En este trabajo, el sistema que se usa es un sistema estéreo, por lo que existen dos conjuntos de parámetros, uno que relaciona las coordenadas de ambas cámaras y que no depende de la escena capturada, y el segundo componente que relaciona el sistema de coordenadas para cada cámara con el sistema de coordenadas de la escena. Este segundo componente definido para cada cámara es independiente de una cámara a la otra.

El proceso que obtiene los parámetros geométricos relacionados con las posiciones de las cámaras en el espacio se llama calibración estéreo. Durante la revisión de la bibliografía, se puede encontrar comúnmente el método de Tsai propuesto en [24]. Otro método (de Haralick) para la calibración es presentado en [12]. De acuerdo a [13], el método de Haralick [12] presenta mejores resultados que el método de Tsai [24] y que el método DLT [1].

Para el desarrollo de este trabajo, se usará un método mejorado y reducido del método de Haralick, el cual es presentado en [13]. Dicho método es descrito en la sección 2.3.6.

La justificación de la calibración reside en que permite simplificar el problema de la correspondencia al lograr que las imágenes estéreo estén rectificadas. Un par de imágenes estéreo se encuentran rectificadas si para cada píxel en la fila *x* de la imagen izquierda, su correspondiente píxel de la imagen derecha se encuentra en la misma fila *x*. Si se cuenta con imágenes estéreo rectificadas, la búsqueda de correspondencias se realiza sólo sobre la línea epipolar, es decir sobre la misma fila.

Espacios tridimensionales.

Sea P_w un punto en el espacio, el cual es representado por el vector columna $P_w = (x_w, y_w, z_w)^T$. Por simplicidad, se usarán coordenadas homogéneas, esto es $P_w = (x_w, y_w, z_w, 1)^T$. Sea P_i la proyección del punto P_w representado en el sistema de referencia de la cámara, entonces P_i se define de la siguiente manera:

$$P_i = T_{wi} \cdot P_w \tag{2.4}$$

Donde T_{wi} es la matriz que transforma las coordenadas del espacio a las coordenadas de la cámara. Esta matriz está formada por la matriz de rotación R_{3x3}, el vector de traslación T_{3x1} y por el coeficiente *K* que modela la distorsión radial. Por lo tanto podemos reescribir la ecuación 2.4 como sigue:

$$P_i = k \cdot \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0 & 1 \end{bmatrix} \cdot P_w \tag{2.5}$$

Imágenes estéreo.

Al observar un objeto o escena desde dos puntos de vista, se puede notar diferencias de una imagen a la otra, estas diferencias son el desplazamiento causado por la distancia a la que se encuentra una cámara de la otra. Otra causa de la diferencia entre las imágenes es generada por el ángulo en que se encuentren las cámaras. En realidad, estos dos factores se encuentran relacionados. A la geometría en la cual la distancia de una cámara a la otra (de un centro óptico al otro) es grande, se le conoce como geometría estéreo de ángulo amplio. Por otro lado, si la distancia es corta, se le conoce como geometría de ángulo estrecho. La geometría estéreo de ángulo amplio entrega mejores estimaciones tridimensionales de la escena capturada, sin embargo, esta geometría presenta diversas limitantes, principalmente que es más difícil establecer la correspondencia en sus imágenes ya que el rango de búsqueda es mucho mayor que el de las imágenes generadas con un sistema estéreo de ángulo estrecho. Otra limitante de la geometría de ángulo amplio es que existe menos traslape entre los campos de visión por lo que existe mayor posibilidad de oclusión.

La figura 2.6 muestra un ejemplo de imágenes estéreo con cierto desplazamiento y no rectificadas. La mayoría de los sistemas estéreo usan configuraciones paralelas. El ser humano cuenta con un sistema visual un tanto distinto, ya que tiene la capacidad de ajustar el ángulo de convergencia, que en las configuraciones paralelas es de 0°. En [11] reportan que el ojo humano puede comúnmente ajustar su distancia focal de 14mm hasta 17mm siendo el desplazamiento entre los ojos aproximadamente 65mm. Este desplazamiento se le conoce como distancia intraocular.

Figura 2.6 Ejemplo de imágenes estéreo

2.3.5 Geometría epipolar

En un sistema estéreo, la principal característica geométrica a la que se debe llegar se le conoce como *la geometría epipolar*. Esta característica permite establecer una correspondencia unidimensional entre las imágenes estéreo. Esto es, al aplicar ciertas transformaciones, se logra que cada píxel en la imagen izquierda tenga su correspondiente píxel ubicado sobre el mismo renglón de la imagen derecha. La figura 2.7 muestra la geometría epipolar.

Si se define a *C (C')* como el centro óptico de la cámara izquierda (derecha). Existe una línea $\overline{CC'}$ llamada línea epipolar que se proyecta al plano *R (R')* hacia los puntos *e (e')* a los cuales se les llama epipolos. Si cumple lo anterior, se dice que la geometría estéreo cumple la restricción epipolar, por lo que un punto *M* que se proyecta en *m* tiene su correspondiente sobre la línea l_m, por lo tanto, la búsqueda de su correspondiente se realiza sólo sobre la línea l_m.

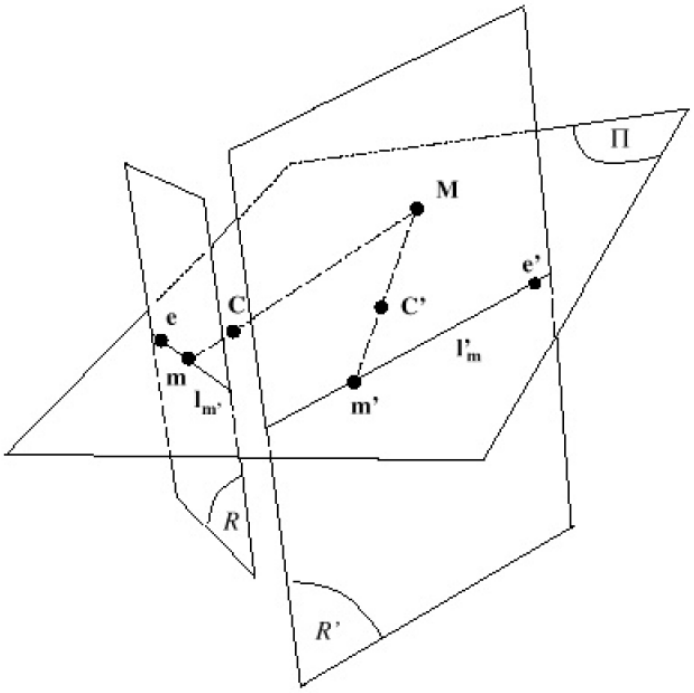

Figura 2.7 Geometría epipolar. Si la línea $l_{m'}$ es igual a la línea l'_m entonces la restricción epipolar se cumple.

La figura 2.8 muestra la línea *T* a la que se le conoce como línea base, el plano (M,C,C') es llamado plano epipolar. Se muestra como los planos de las imágenes donde se proyecta el punto *M* pueden ser modificados para que cumplan con la restricción epipolar. Los planos modificados se muestran en líneas punteadas y son llamados planos rectificados,

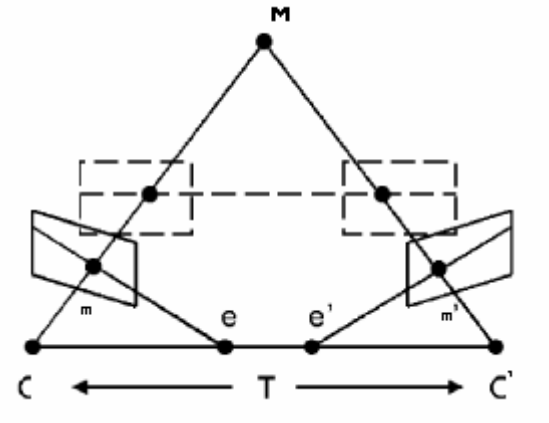

Figura 2.8 Imágenes rectificadas.

Para cumplir con la restricción epipolar, se puede montar el sistema estéreo con los planos de las imágenes de tal manera que sean coplanares y que tengan la misma línea base. Sin embargo, la precisión del montaje mecánico debería ser sub-milimétrica, por lo que su implementación es poco práctica. Existen soluciones analíticas para alcanzar la restricción epipolar, éstas serán discutidas en las siguientes secciones.

2.3.6 Calibración

En la sección 2.3.4 se describió como un punto en el espacio se proyecta en el plano de la imagen a través de una representación matricial. La ecuación 2.5 hace uso de la matriz de transformación T_{wi} la cual se define a través de las matrices de rotación y traslación así como por los coeficientes de distorsión radial. Para poder conocer con precisión las coordenadas de cualquier punto en el espacio proyectado en el plano de la imagen, es necesario contar con el conocimiento de varios parámetros. Estos parámetros se dividen categóricamente en dos grupos, siendo estos los parámetros extrínsecos e intrínsecos.

Los parámetros extrínsecos son aquellos que definen la posición y orientación de la cámara con respecto al sistema de referencia externo, mientras que los parámetros intrínsecos son aquellos que definen la geometría interna de la cámara. Al proceso por el cual se obtiene el conocimiento de estos parámetros se le conoce como ***calibración***.

En la tabla 2.1 se enlistan los parámetros geométricos de la cámara.

Parámetros extrínsecos.	Parámetros intrínsecos
Matriz de rotación [3x3] R	Distancia focal f
Vector de traslación [3x1] T	Tamaño del píxel S_x, S_y
	Coordenadas del centro de la imagen C_x,C_y
	Coeficientes de distorsión radial α_1, α_2, α_3.

Tabla 2.1 Parámetros intrínsecos y extrínsecos.

Existen diversos métodos para realizar la calibración, entre ellos, el método de Haralick [12] y el método de Tsai [24]. Otro método es el de Abdel [1] llamado DLT y que es uno de los más usados en visión computacional. Sin embargo, en [13] se evalúan los tres métodos anteriores mostrando que el método de Tsai produce los peores resultados. En [13] se propone el método de Haralick con una reducción en su costo computacional, el cual entrega mejores resultados que los encontrados por DLT, Tsai y Haralick. Debido a esto, el método de calibración que se usó en este trabajo fue el de Haralick reducido [13]. La simplificación al método de Haralick descrita en [13] se realiza al introducir un mapeo lineal al sistema de transformación de las coordenadas del espacio tridimensional a las coordenadas del plano de la imagen. Los parámetros de calibración son generados al resolver un sistema de ecuaciones lineales sobreestimado a través del método de mínimos cuadrados. Otro aporte en su trabajo fue el de manejar la distorsión radial de la lente a través de un concepto al cual ellos llaman “Calibración localizada” en el cual se divide la imagen en 9 ventanas, donde cada ventana tiene sus propios valores para la distorsión radial. De acuerdo a sus conclusiones, la técnica presentada mejora significativamente los resultados de la calibración con respecto a los métodos Haralick, Tsai y DLT.

2.4 Correspondencia Estéreo (Taxonomía de métodos)

Dado un par de imágenes estéreo, se puede obtener de ellas un mapa de disparidades. Estos mapas se pueden obtener a través de distintos métodos y cumpliendo distintas restricciones. A estos métodos se le conoce como métodos de correspondencia estéreo y su objetivo principal es el de correlacionar a un píxel de una de las imágenes con su correspondiente en la otra imagen. Comúnmente [16] estos métodos de correspondencia se dividen principalmente en dos grupos, locales y globales. En los métodos locales, la comparación se realiza sobre pequeñas ventanas de una imagen con otras ventanas en la otra imagen. Los métodos globales complementan a los anteriores al realizar la comparación en líneas completas o incluso en toda la imagen.

Los métodos locales son sensibles a regiones con ambigüedades, por ejemplo a regiones ocluidas. Los métodos globales son más robustos contra estos problemas, sin embargo, son mucho más costosos en tiempos computacionales.

Los métodos locales realizan comparaciones de ventanas para encontrar una métrica que especifica la similitud entre las ventanas. Las métricas de comparación se calculan a través de funciones llamadas **medidas de similitud**, las cuales se basan en operaciones matemáticas simples (restas, sumas) con números de tipo entero. La implementación en hardware de operaciones matemáticas que no requieren el uso de números de punto fijo o flotante, requieren de menor consumo de recursos, también, estas implementaciones reducen el tiempo de procesamiento. En las siguientes secciones se

Métodos	Referencia	Descripción
Métodos globales		
Programación dinámica	(Quénot, 1992)	Realiza una división de las imágenes al agrupar varios renglones, después, a estos los divide en vectores y a ellos los compara con los de la otra imagen.
Curvas intrínsecas	(Tomasi et al, 1995, 1998)	Convierte las líneas epipolares a curvas intrínsecas y la solución se encuentra resolviendo el problema del vecino más cercano.
Grafos de corte	(Boykov et al, 2001)	Encuentra la disparidad al seleccionar el corte mínimo del flujo máximo del grafo.
Difusión no lineal	(Mansouri et al, 1998)	Se aplica el proceso de difusión local.
Propagación de probabilidades	(Sun et al, 2002)	Se resuelve usando una red de probabilidades comunicada con paso de mensajes.
Métodos locales		
Correlación por áreas	(Fua, 1991) (Parr et al., 1976) (Zabih et al. 1997)	Se comparan regiones o ventanas de la imagen mediante medidas de similitud.
Basados en el gradiente	(Lucas et al., 1981)	Se minimiza una función sobre una pequeña ventana.
Correlación por características	(Faugeras et al., 1998) (Fua et al., 1995)	Se comparan características de las imágenes como esquinas o círculos.

Tabla 2.2 Métodos de correspondencia

hará una breve descripción de los métodos de correspondencia globales y locales con particular interés en la correlación por área. En [16] se realiza una revisión a los métodos de correspondencia estéreo. La tabla 2.2 resume algunos de los métodos globales y locales descritos por Myron [16].

2.4.1 Métodos locales

Los métodos de correspondencia estéreo local realizan la comparación por ventanas, esto es, dada una ventana de dimensión $N_w \, x \, M_w$ en una imagen, se busca su correspondiente en la otra imagen. Estos métodos se subdividen comúnmente en tres distintas categorías: métodos basados en el Gradiente, correlación basada en características y correlación basada en áreas.

Métodos basados en el Gradiente.

Para hallar la disparidad de un punto dado en la escena capturada por un sistema estéreo se puede plantear como una ecuación diferencial la cual relaciona el movimiento y la intensidad de brillo de dicho punto. Esta manera de calcular la disparidad se conoce como método basada en el gradiente o en el flujo óptico. Sin embargo, este método se basa en la restricción de asumir que el brillo del punto es el mismo en ambas imágenes, la cual no siempre se cumple debido a las características de los sensores del sistema estéreo. Otra razón de por que la intensidad de un punto no siempre es la misma en ambas imágenes se debe a la posición de los sensores, ya que factores de luz como sombras, afectan la captura de los rayos de luz.

Sin embargo, asumiendo que la intensidad de los puntos son los mismos en ambas imágenes, la traslación de la proyección del punto en una imagen hacia la otra se puede modelar mediante la siguiente ecuación.

$$(\lambda_x E)v + E_t = 0 \tag{2.6}$$

Donde $\lambda_x E$ es el valor del gradiente en la horizontal de la imagen y E_t es la derivada temporal, es decir la diferencia entre las dos imágenes. v es la traslación entre las dos imágenes. La complejidad de realizar la correspondencia entre las dos imágenes usando flujo óptico es *O(N)*, con *N* número de píxeles en las imágenes estéreo.

La disparidad de cualquier punto se puede calcular mediante la solución al sistema de ecuaciones diferenciales lineales usando el método de mínimos cuadrados. El sistema se define como:

$$v = \left(A^T A\right)^{-1} A^T b \tag{2.7}$$

donde

$$A = \begin{bmatrix} \lambda_x E(p1) \\ \lambda_x E(p2) \\ \vdots \\ \lambda_x E(p_{nxn}) \end{bmatrix} \text{ y } b = -\begin{bmatrix} E(p1) \\ E(p2) \\ \vdots \\ E(p_{nxn}) \end{bmatrix} \tag{2.8}$$

Ahora, la complejidad para resolver el mapa de disparidad es *O(Nn)*, donde *n* es el número de píxeles de la ventana a computar. *O(Nn)* es comparable con realizar una búsqueda exhaustiva dentro de las imágenes.

Correlación basada en características.

El principal aporte de los métodos de correspondencia basados en características es que son robustos ante oclusión, escenarios con textura uniforme y ante la discontinuidad de profundidad. Los anteriores son problemas que los métodos basados en el gradiente así como los métodos de correspondencia basados en áreas son sensibles.

La razón de la robustez de los métodos basados en características se debe a que la correlación que realizan es sobre características bien definidas en las imágenes estéreo, estas pueden ser bordes, esquinas, círculos, etc. Sin embargo, un preprocesamiento es requerido para detectar estas características. Existen diversos métodos para la detección de características como SUSAN [22]. Debido a las restricciones anteriores, la densidad del mapa de disparidades se ve limitado. Debido a que la mayoría de las aplicaciones necesitan contar con mapas de disparidad densos, el interés en los métodos de correspondencia basados en características ha ido decayendo durante la última década [16].

Una vez detectadas las características, se procede a realizar la búsqueda de correspondencias. En [25] se propone usar un sistema de jerarquías para resolver el problema de correspondencia. El método de Venkateswar se resume en los siguientes pasos:

1. Se extraen las características (En particular, bordes).
2. Se construye un grafo jerárquico con las características encontradas.

La construcción del grafo se basa en relaciones estructurales y preceptúales. Las relaciones incompatibles también son usadas con el fin de reforzar el agrupamiento de características.

3. Una vez terminado el grafo, se podan las características incompatibles.
4. Se realiza la correlación por características usando los grafos de cada imagen estéreo, iniciando con la raíz y llevando un flujo TOP-DOWN. Una vez realizada la correlación con el nivel superior, las características contenidas en este nivel ya no son necesarias en las siguientes búsquedas en los niveles inferiores.

Se observa del paso 4, que el espacio de búsqueda se reduce significativamente por cada nivel que se finaliza. Los autores de [25] reportan que la complejidad del problema de correspondencia para cada nivel es $O(N^4)$ donde N es el número de características en el nivel. Nótese que estos métodos solo obtienen disparidades (por lo tanto información de profundidad) en las características encontradas, por lo tanto, no producen mapas de disparidades densos.

Correlación basada en áreas.

El objetivo de los métodos de correspondencia basados en áreas es estimar la disparidad de un punto en una imagen al comparar la ventana de dimensiones *nxm* con una seria de ventanas de la otra imagen. Como se estableció en la sección 2.3.5, si el par de imágenes estéreo se encuentran rectificadas, la búsqueda se reduce a una dimensión. La búsqueda de correspondencia se basa en estimar la similitud de una ventana de una imagen con la otra ventana de la otra imagen. Para calcular esta semejanza se utilizan medidas de similitud, las cuales comúnmente se dividen en tres clases: me-

didas de correlación, medidas de diferencias de intensidad y métricas de rango.

La medida de similitud llamada Correlación Cruzada Normalizada (NCC por sus iniciales en inglés) es la medida estándar estadísticamente más usada. Por otro lado, la medida de similitud basada en la suma de las diferencias cuadradas (SSD) es un método computacional más simple que el de la correlación cruzada. Por último, existe otra medida de similitud, la cual se basa en la suma absoluta de las diferencias (SAD), y esta ha sido encontrada como la medida de similitud que se usa con frecuencia por su eficacia computacional [7]. En [23] se reporta que la medida de similitud SSD *es usualmente preferible que* la SAD debido a que la SSD no es sensible a la modificación causada por regiones con valores de intensidad muy pequeños o muy grandes.

La clase de medidas de similitud basada en métricas de rango fueron propuestas en [32]. Las medidas hacen uso de la relación que existe en el píxel central con los píxeles en su vecindad contenidos en su ventana. Estas medidas se basan en la transformada Census y Rank. La transformada Census relaciona a cada píxel y su vecindad con un vector de elementos boléanos. A dicho vector se le conoce como vector Census. Una vez determinados los vectores Census, se pueden comparar usando la diferencia entre ellos, la cual puede medirse el número de elementos que difieren en cada uno usando la distancia Hamming.

La transformada Rank realiza los mínimos pasos que la transformada Census, a diferencia, que cuando ya se obtuvieron los vectores Census, se suman los elementos de cada vector. El cálculo de la similitud, ahora se puede resolver con SAD o SSD. En [4] se compara el rendimiento de las métricas anteriores y entregan como resultado que medidas de similitud basadas

en métricas de rango son mas robustas ante la distorsión radiométrica y oclusión. Sin embargo, las transformadas Rank y Census incrementa la dimensión de los datos de la imagen, haciéndolos caros en sus recursos de cómputo.

La tabla 2.3 resume las medidas de similitud descritas anteriormente. La figura 2.9 describe los términos usados en la tabla 2.3

Medida de similitud	Definición
Correlación cruzada normalizada NCC	$\frac{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)\cdot\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2\cdot\left(I_2(u+d,v)-\bar{I}_2\right)^2}}$
Suma de diferencias cuadradas SSD	$\sum_{u,v}\left(I_1(u,v)-I_2(u+d,v)\right)^2$
SSD Normalizada	$\sum_{u,v}\left(\frac{\left(I_1(u,v)-\bar{I}_1\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2}}-\frac{\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_2(u+d,v)-\bar{I}_2\right)^2}}\right)^2$
Suma absoluta de diferencias	$\sum_{u,v}\left\|I_1(u,v)-I_2(u+d,v)\right\|$
Transformada Rank	$\sum_{u,v}\left(I_1'(u,v)-I_2'(u+d,v)\right)$ $I_k'(u,v)=\sum_{m,n}I_k(m,n)<I_k(u,v)$
Transformada Census	$\sum_{u,v}HAMMING\left(I_1'(u,v),I_2'(u+d,v)\right)$ $I_k'(u,v)=BITSTRING_{m,n}\left(I_k(m,n)<I_k(u,v)\right)$

Tabla 2.3 Medidas de similitud

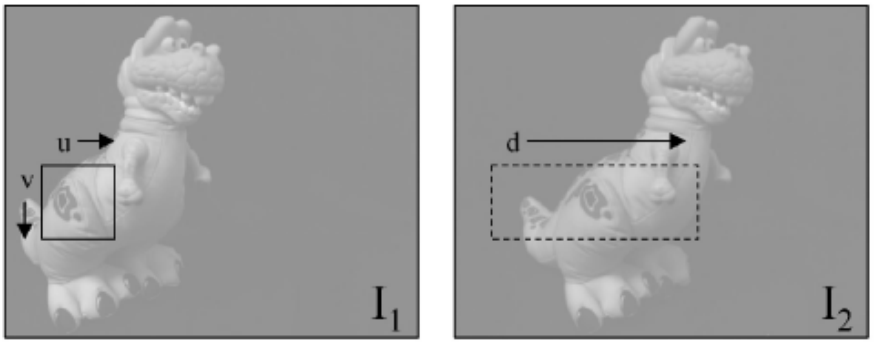

Figura 2.9 Términos de la búsqueda de correspondencias.

Las medidas de similitud hasta ahora analizadas trabajan sobre imágenes codificadas en niveles de gris. Sin embargo, la extensión de las medidas de similitud a imágenes en color ha sido planteada por Chambon en [8] en donde se muestra cómo la correlación puede ser usada con imágenes con color. De igual manera, se presenta un protocolo que permite seleccionar el espacio de colores adecuado para la correlación, así como los métodos que fusionan los canales de color para su aplicación con una medida de similitud. Los resultados reportados demuestran que la correlación estéreo puede ser mejorada usando imágenes a color.

Chambon et al [8] concluyen demostrando que usando cualquiera de los dos métodos propuestos por ellos y usando el espacio de color XYZ se generan los mejores resultados. En la sección 3.4.2 se describe con mayor detalle los métodos propuestos en [8].

Esta sección se finaliza mostrando los pasos típicos necesarios para generar el mapa de disparidad denso de una escena capturada con un sistema estéreo.

1. *Corrección geométrica.*
    a. *Calibración estéreo.* Encontrar los parámetros intrínsecos y extrínsecos del sistema estéreo.
    b. *Geometría epipolar.* Establecer la geometría que permita establecer líneas epipolares en las imágenes estéreo.
    c. *Rectificación estéreo.* Transformar las imágenes para que cumplan la geometría epipolar.
2. *Transformación de intensidad.* Si es necesario, se puede aplicar alguna transformación en las imágenes para normalizar sus intensidades. Por ejemplo, ecualizar las imágenes estéreo.

3. *Solucionar el problema de correspondencia.* Utilizar algún método de apareamiento de píxeles.
4. *Creación de mapa de disparidad denso.* Determinar los valores óptimos de las medidas de similitud para encontrar la disparidad de cada punto en las imágenes estéreo.

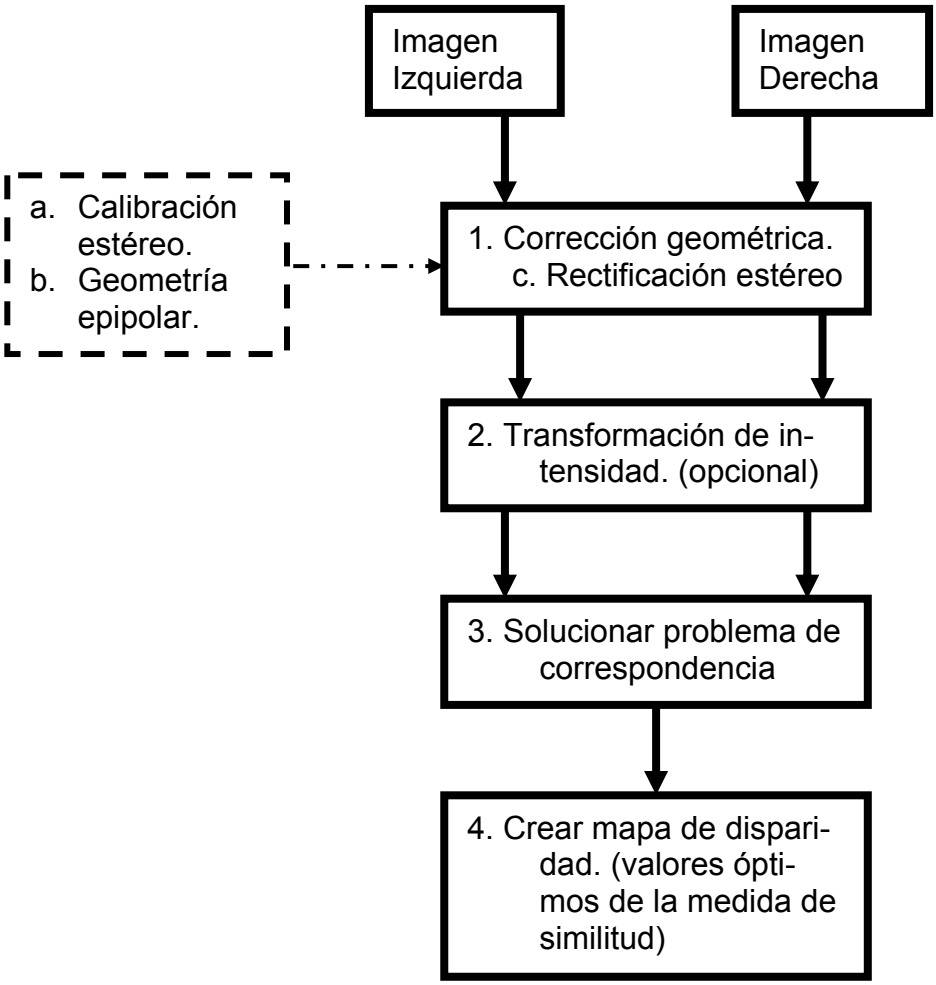

Figura 2.10 Diagrama de flujo para generar mapa de disparidad

2.4.2 Métodos globales

Los métodos globales buscan reducir los errores producidos por correspondencias falsas en el par de imágenes estéreo. Para reducir estos

errores, los métodos globales se basan en restricciones no locales, es decir, la búsqueda de correspondencias se realiza sobre líneas completas del par estéreo o incluso en toda la imagen. Los métodos globales reducen los errores en la estimación de correspondencia cuando las imágenes a procesar tienen ciertas características, por ejemplo, imágenes con oclusiones, texturas uniformes, entre otras. Sin embargo, el costo computacional de los métodos globales es significativamente mayor al costo de los métodos locales. Debido a lo anterior, en este trabajo se propone una arquitectura hardware basada en FPGA para la recuperación tridimensional usando métodos locales, particularmente en los métodos basados en la correlación por área.

2.5 Arquitecturas Hardware de visión estéreo.

Las arquitecturas hardware de visión estéreo han tenido un gran auge durante la última década [16], principalmente debido a la gran cantidad de aplicaciones que requieren de la estimación de la estructura tridimensional de los objetos que analizan, como pueden ser sistemas médicos, realidad virtual, sistemas de monitoreo industrial, entre otras.

Sin embargo, los algoritmos para generar un mapa de disparidad (figura 2.9), requieren de un alto poder de cómputo para una implementación a velocidad de video (30 cuadros por segundo) debido a la gran cantidad de operaciones iterativas. Una implementación directa en software impone la restricción de ejecutar los algoritmos de manera secuencial, por lo que no es factible alcanzar restricciones de tiempo real. El orden para calcular el mapa de disparidad de un par de imágenes estéreo de N número de píxeles con una ventana de búsqueda de n píxeles en un rango de disparidad l es de $O(Nnl)$.

Por otro lado, si se realiza un análisis a los algoritmos que resuelven los problemas de la visión estéreo, se puede identificar paralelismo de datos, por lo tanto, identificar instrucciones que son independientes entre ellas con respecto al tiempo. Identificar instrucciones independientes con respecto al tiempo, permite proponer una arquitectura capaz de ejecutar dichas instrucciones en el mismo tiempo logrando que la arquitectura sea paralela.

Debido a las arquitecturas paralelas, se puede encontrar en la revisión del avance científico diversas arquitecturas hardware basadas en arreglos de compuertas programables en el campo (FPGAs) y procesadores digitales de señales (DSP) entre otras plataformas. Restricciones en la velocidad de procesamiento se establecen para cada arquitectura, siendo el objetivo de todas ellas alcanzar la velocidad de video, 30 cuadros por segundo (*fps*). Myron et al presentan en [16] una recapitulación de las arquitecturas hardware desarrolladas durante la década pasada. Durante nuestra investigación, se han encontrado más arquitecturas, con las cuales se complementa la recapitulación en [16].

En 1993, se reporta en [9] dos arquitecturas hardware, una basada en DSP y otra en FPGA que implementan la correlación cruzada normalizada. De igual manera se implementó el método NCC en una SPARC 2 bajo lenguaje C. Los resultados muestran que a pesar de la dificultad de la implementación en el FPGA, ésta supera en un factor de 34 veces con respecto a la velocidad de procesamiento a la implementación en DSP y en 210 veces a la implementación en software.

En 1997, se desarrolló otra arquitectura basada en FPGAs [28]. Woodfill et al. [28] presentan su arquitectura en la tarjeta PARTS engine, la cual se basa en 16 FPGAs Xilinx 4025 montados en una tarjeta PCI. Los resultados que reportan es velocidad de procesamiento de 42 cuadros por segundo tra-

bajando con imágenes de 320x240 píxeles codificadas en niveles de gris. La implementación de la medida de similitud fue la trasformación Census.

En 2000, Xicotencatl [30] propone una arquitectura estéreo basada en FPGA usando como medida de similitud la suma absoluta de diferencias (SAD), reportando como resultados velocidad de procesamiento de 85 fps trabajando con imágenes de 240x240.

En [16] se reporta una implementación comercial llamada Point Grey Triclops creada en 2001. Esta implementación esta basada en un máquina con procesador Pentium IV a 1.4 GHz. Los resultados entregados es velocidad de procesamiento de 13 fps. sobre imágenes de 320x240 píxeles. Este sistema usa 3 cámaras.

En 2003, Ahmad [2] desarrolló un sistema estéreo basado en un conjunto de 4 FPGAs usando como medida de similitud LWPC (Local Weighted Phase Correlation). Sus resultados reportan una velocidad de procesamiento de 30 fps usando imágenes de 256x360 píxeles.

En Noviembre de 2006, la empresa Point Grey Research, presenta su producto comercial *Bumblebee XB3* [19] el cual trabaja con imágenes de 1280x960 a una velocidad de procesamiento de 15 fps. *Bumblebee* está basado en un máquina con procesador Pentium IV a 2 GHz.

En Mayo de 2007, la empresa Tyzx, presenta su producto comercial *DeepSea G2 Stereo Vision* [27], desarrollado por Woodfill et al. Este sistema trabaja con imágenes capturadas por sensores de la marca National, de tamaño 512x480 y su velocidad de procesamiento es de 30 fps.

Arquitectura HW	Tamaño de la imagen	Velocidad de procesamiento	Medida de simili-tud	Procesador	Cámaras	#píxeles por se-gundo
INRIA 1993	256x256	3.6 fps	NCC	Perle-1	3	235,930
CMU iWarp 1993	256x240	15 fps	SSAD	64Processor iWarp2	3	921,600
Teleos 1995	320x240	0.5 fps	SC	Pentium 166 Mh6z	2	38,400
JPL 1995	256x240	1.7 fps	SSD	Datacube	2	104,448
CMU 1995	256x240	30 fps	SSAD	Custom HW y DSP	6	1,843,200
Point Grey Triclops 1997	320x240	6 fps	SAD	PII 450 MHz.	3	460,800
SRI SVS 1997	320x240	12 fps	SAD	TSM3920 DSP	2	921,600
PARTS engi-ne 1997	320x240	42 fps	Census	FPGA	2	3,225,600
CSIRO 1997	256x256	30 fps	Census	FPGA	2	1,966,080
SAZAN 1999	320x240	20 fps	SSAD	FPGA	9	1,536,000
Point Grey Triclops 2001	320x240	13 fps	SAD	PIV 1.4GHz.	3	998,400
SRI SVS 2001	320x240	30 fps	SAD	PII 700MHz.	2	2,304,000
Xicotencatl. 2000 [30]	240x240	85 fps	SAD	FPGA	2	4,896,000
VRSV 03 [2]	256x360	30 fps	LWPC	4 FPGAs	2	2,764,800
Bumblebee XB3 2006 [19]	1280x960	15 fps	SAD	PIV 2.0GHz. 512RAM 64AGP	2	18,432,000
DeepSea G2 Stereo Vision 2007 [27]	512x480	30 fps	Census	FPGA, DSP, PowerPC, DeepSea	2	7,372,800

Tabla 2.4 Arquitecturas Hardware de visión estéreo.

La tabla 2.4 muestra la recopilación creada por Myron et al [16], sobre las arquitecturas hardware de visión estéreo, actualizada con las arquitecturas encontradas en la literatura hasta la fecha.

La última columna de la tabla 2.4 muestra el número de píxeles procesados por cada una de las arquitecturas hardware. Este número es una métrica para comparar la velocidad de procesamiento entre las distintas arquitecturas.

2.6 Discusión.

Durante este capítulo se han revisado los numerosos intentos por desarrollar sistemas capaces de entregar descripciones tridimensionales de objetos capturados por sistemas estéreo. Se observa que el intenso esfuerzo por desarrollar arquitecturas implementadas en sistemas de cómputo de propósito específico tienen su justificación en el hecho de contar con arquitecturas paralelas que sean capaces ejecutar instrucciones con datos independientes con respecto al tiempo.

Debido a la naturaleza de los algoritmos de visión, existe una gran cantidad de operaciones iterativas sobre los datos de las imágenes a procesar. Las operaciones son, en la mayoría de los algoritmos de visión, de tipo matricial, lo que permite analizar los algoritmos para identificar instrucciones independientes con respecto al tiempo. La ejecución de instrucciones de forma paralela, resuelve el problema de los sistemas de cómputo de propósito general: la ejecución secuencial. Sin embargo, ésta no es la única limitante del procesamiento de los algoritmos de visión (o cualquier otro tipo de algoritmo). Las memorias que almacenan las imágenes tienen como características físicas el acceso restringido a un dato por ciclo de reloj. Es necesario

establecer métodos que permitan explotar el ancho de banda de las memorias, con el fin de reducir el número de accesos y aumentar la velocidad de desempeño del sistema.

La revisión del avance científico permite establecer que a pesar de las múltiples arquitecturas estéreo, la limitante más grande de éstas es el compromiso velocidad de procesamiento contra el tamaño de imágenes a procesar. Existen arquitecturas que alcanzan 30 fps pero usan imágenes con resolución menor a la resolución VGA (640x480) [2,27]. Por otro lado, las arquitecturas que son capaces de procesar imágenes de resolución mayor a la resolución VGA no alcanzan la velocidad de procesamiento de 30 fps [19].

La implementación del algoritmo estéreo en una arquitectura basada en FPGA, permite diseñar un sistema paralelo y que pueda establecer métodos para aumentar el ancho de banda de las memorias, usando componentes de almacenamiento a diferentes niveles (memoria RAM, bloques RAM y registros internos del FPGA). Si el diseño del sistema estéreo es modular, permitiría una implementación en una arquitectura hardware basada en FPGA escalable, es decir, agregar tantos módulos como sean necesarios para mejorar el desempeño del sistema estéreo.

En el siguiente capítulo, se muestra la arquitectura estéreo propuesta en esta investigación. Dicha arquitectura explota las condiciones antes mencionadas acerca del paralelismo, además, se ha explorado e implementado la capacidad de aumentar el ancho de banda de las memorias, reduciendo el número de accesos a la memoria.

Capítulo 3

Arquitectura Hardware para la recuperación 3D.

3.1 Introducción.

Los algoritmos estéreo requieren de un alto poder de cómputo para alcanzar la principal restricción, la velocidad de video (30 cuadros por segundo). La implementación de estos algoritmos en sistemas de cómputo de propósito general tiene como limitante la ejecución de instrucciones de forma secuencial, por lo que difícilmente, estas implementaciones alcanzan la restricción de velocidad de video.

Sin embargo, a pesar de contar con una arquitectura capaz de ejecutar varias instrucciones en paralelo, existe la limitante de acceso a los datos debida a las características físicas de las memorias que almacenan las imágenes, es decir, sólo se puede acceder a un solo dato (en algunos casos, hasta dos datos) de la memoria en un ciclo de reloj. Por lo tanto, es necesario explotar el ancho de banda de estas memorias para reducir el número de accesos a la misma. Otra técnica es implementar bloques de almacenamiento (*buffers*) dentro de la arquitectura. Ambas técnicas han sido exploradas en esta investigación.

En el resto del capítulo, se describe la arquitectura hardware basada en FPGA que entrega un mapa de disparidad dado un par de imágenes estéreo sin rectificar. A continuación se describe de manera general la arquitectura, así como cada módulo funcional dentro de la arquitectura.

3.2 Descripción general de la arquitectura.

La arquitectura debe ser capaz de recibir el flujo de imágenes estéreo, realizar sobre ellas transformaciones geométricas con el fin de alcanzar la restricción epipolar, para luego ser procesadas para obtener un mapa de disparidades denso, usando el algoritmo de correlación estéreo basado en área bajo la medida de similitud SAD con el criterio del mínimo.

Usando la metodología TOP-DOWN, se presenta el diagrama de bloques de la arquitectura hardware basa en FPGA para la reconstrucción 3D en tiempo real. Primero se describe la arquitectura global cuyo funcionamiento ya ha sido planteado, para luego presentar el diagrama a bloques y el funcionamiento de los módulos individuales.

En la figura 3.1 se muestra el diagrama a bloques de la arquitectura global propuesta. Los módulos que se encuentran agrupados por un margen de línea punteada, corresponden a la arquitectura presentada en este trabajo, mientras que los módulos que se encuentran en su exterior, son módulos periféricos propuestos para la comunicación con la arquitectura realizada. A continuación, se describe el funcionamiento de la arquitectura, así como el funcionamiento de cada módulo.

La arquitectura guarda en las memorias RAM1 y RAM2, la imagen izquierda y derecha, respectivamente. El procesamiento inicia cuando el primer píxel de cada imagen es ingresado a la arquitectura, este píxel y los demás, van siendo colocados en las coordenadas corregidas por el módulo *Transformación Geométrica* para entregar un par estéreo que cumple con la restricción epipolar.

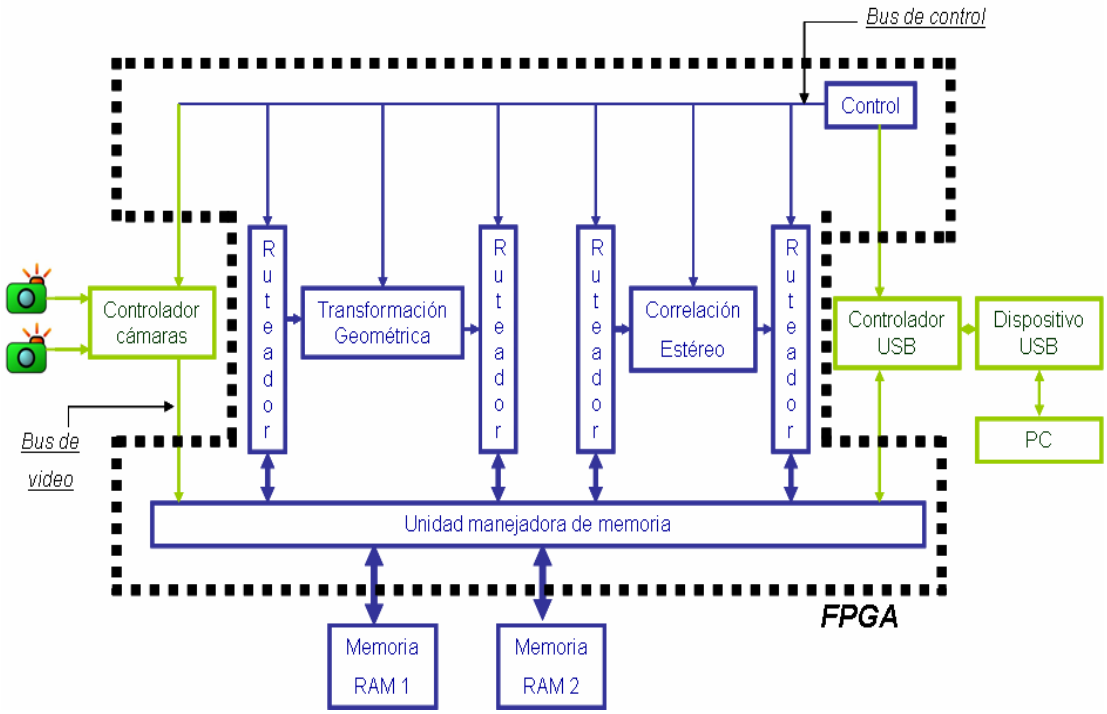

Figura 3.1 Diagrama a bloques: Arquitectura global propuesta.

El módulo *Correlación estéreo* realiza el algoritmo de correlación sobre las imágenes rectificadas aplicando como medida de similitud la SAD y como criterio de selección al mínimo. El módulo *Correlación estéreo* entrega el mapa de disparidad que contiene los índices en donde se genera el mínimo valor de la SAD (este valor maximiza la similitud de dos ventanas).

El módulo *Unidad manejadora de memoria* accede a las RAM para escribir los píxeles provenientes de las cámaras estéreo, para que después los módulos *Ruteadores* puedan leer los datos para ser accedidos por los módulos *Transformación geométrica y Correlación estéreo*. A pesar de que los módulos *Ruteadores* tienen el mismo fin, no tienen la misma funcionalidad,

ya que, como se verá después, los módulos de procesamiento requieren datos de la memoria en diferentes tiempos así como de diferentes localidades.

Todos los módulos ya descritos son controlados y sincronizados por el módulo *Control*, el cual se encarga de habilitar y deshabilitar los demás módulos con el fin de llevar el flujo de datos correspondiente, así como de configurar los módulos para el funcionamiento explicado. El módulo *Control* es la máquina de estados de la arquitectura estéreo.

3.3 Descripción de los módulos funcionales.

En la sección anterior se describió de manera general, el funcionamiento de la arquitectura hardware para la recuperación 3D, ahora se analiza el funcionamiento individual de cada módulo funcional de la arquitectura.

3.3.1 Módulo Transformación Geométrica.

El principal objetivo del módulo *Transformación Geométrica* es procesar las imágenes de entrada del sistema estéreo para generar imágenes que cumplan con la restricción epipolar.

Como fue discutido en la sección 2.3.3, una manera de acelerar los algoritmos de correcciones geométricas que eliminan diversas distorsiones (distorsión radial, distorsión de proyección) es a través del uso de tablas de mapeo. La idea es tomar un píxel con coordenadas distorsionadas x_d, y_d y reubicarlos en las coordenadas corregidas x, y (véase ecuación 2.3). Las tablas de mapeo contienen las coordenadas correctas para las coordenadas distorsionadas respectivas (véase figura 2.5). El cálculo de los valores de las tablas de mapeo es un proceso que se puede realizar fuera de línea (*off-line*),

es decir, no es necesario realizar el cálculo del modelo matemático que elimina la distorsión para cada píxel. El cálculo de las coordenadas correctas se realiza una vez para cada sensor.

En la sección 2.3.3, se discute la implementación de la corrección radial sobre una imagen distorsionada. Dicha corrección se basa en el modelo matemático descrito por la ecuación 2.3. Se muestra nuevamente la ecuación por comodidad.

$$\begin{aligned} x &= x_d\left(1+\alpha_1 r^2+\alpha_2 r^4+\alpha_3 r^6\right), \\ y &= y_d\left(1+\alpha_1 r^2+\alpha_2 r^4+\alpha_3 r^6\right), \end{aligned} \tag{2.3}$$

La ecuación 2.3, relaciona las coordenadas *x, y* con las coordenadas x_d. y_d mediante una transformación escalar, es decir, las coordenadas x_d. y_d son multiplicadas por un polinomio, en este caso (2.3), de grado seis. El polinomio es definido a través de las variables α_1 (primer coeficiente de distorsión radial), α_2 (primer coeficiente de distorsión radial), α_3 (primer coeficiente de distorsión radial), r (Distancia desde la coordenada hacia el centro de la imagen). Sin embargo, el polinomio de transformación puede ser extendido para contemplar las variables relacionadas con las distorsiones provocadas por la perspectiva de proyección, es decir, la distorsiones de movimiento de traslación y rotación. Dichas distorsiones son definidas en los parámetros extrínsecos de la cámara y son las variables que deben ser agregadas al modelo matemático. El nuevo modelo matemático descrito por el polinomio no lineal de 10 variables se puede consultar en [20]. Por lo tanto, el modelo matemático que modifica las imágenes estéreo para cumplir con la geometría epipolar es el siguiente.

$$\begin{aligned} x &= \left(1 + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6\right) x_t \\ y &= \left(1 + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6\right) y_t \end{aligned} \tag{3.1}$$

Donde x_t, y_t son las coordenadas modificadas por los parámetros de rotación y traslación:

$$\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = R \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} + T \tag{3.2}$$

La ecuación 3.2 hace uso de las matrices *R* y *T*, estas matrices son los parámetros extrínsecos de la cámara que definen la distorsión de movimiento de traslación y rotación respectivamente (véase tabla 2.1).

Siguiendo la idea de las tablas de mapeo, se define el funcionamiento del módulo *Transformación Geométrica*. El módulo debe realizar la transferencia de píxeles desde una coordenada distorsionada hacia otra coordenada corregida, esto se realiza a través de una tabla de mapeo la cual está implementada en un espacio reservado de la memoria RAM.

El flujo de datos inicia en el módulo de control, el cual envía las coordenadas del píxel actual al módulo *Ruteador*, éste lee de la memoria de la imagen (RAM1 o RAM2) el píxel correspondiente y envía las coordenadas (distorsionadas) al módulo *Transformación Geométrica* que busca dentro de la tabla de mapeo que se encuentra en un espacio reservado de la memoria, las coordenadas correctas. A continuación, el módulo *Ruteador* recibe la dirección base correcta y escribe el píxel en las coordenadas correctas dentro la memoria RAM correspondiente.

Las memorias utilizadas en éste trabajo de investigación almacenan datos de hasta 32 bits, por lo tanto almacenan hasta 4 píxeles por localidad, por lo que el módulo *Transformación Geométrica* recibe las coordenadas base, es decir del primer píxel, y entrega la dirección base de los 4 píxeles.

El proceso anterior se realiza para las dos imágenes del par estéreo, por lo tanto, existen dos tablas de mapeo, una para cada imagen y cada tabla de mapeo se encuentra en el espacio reservado de memoria RAM correspondiente. El diagrama a bloques del módulo *Transformación Geométrica* se ilustra en la figura 3.2.

Figura 3.2 Diagrama a bloques: Módulo T*ransformación Geométrica.*

Módulos Ruteadores para Transformación Geométrica.

El funcionamiento del módulo *Transformación Geométrica* es encontrar la nueva dirección en donde debe escribirse el píxel actual para generar imágenes estéreo rectificadas. Por lo tanto, se define la responsabilidad del módulo *Ruteador* para la transformación geométrica, que es la de proveer las direcciones de la tabla de mapeo al módulo *Transformación Geométricas*. El flujo de datos para realizar la transformación geométrica inicia para cada píxel leído, recibir las coordenadas correctas para reescribirlo en la localidad correspondiente. A primera vista, pareciera que el funcionamiento del módulo *Ruteador* podría ser asignado al módulo *Transformación Geométrica*, o viceversa con el fin de ahorrar simplicidad en el diseño y tal vez recursos del dis-

positivo FPGA. Sin embargo, la razón de la implementación del módulo ruteador es permitir posibles extensiones en el funcionamiento del módulo *Transformación Geométrica*, dejando como única función del ruteador, la de direccionar datos al módulo *Transformación Geométrica*. En la sección 3.4.1 se propone una extensión para mejorar el rendimiento de la transformación geométrica al realizar una interpolación bilineal.

El diagrama a bloques correspondiente al módulo *Ruteador* para transformación geométrica se muestra en la figura 3.3.

Figura 3.3 Diagrama a bloques: Módulo Ruteador para transformación geométrica.

3.3.2 Módulo Correlación Estéreo.

El módulo *Correlación Estéreo* es el encargado de realizar el algoritmo de correlación estéreo basado en la medida de similitud SAD usando como criterio de selección al mínimo, es decir, el índice donde ocurre el mínimo valor SAD, dicho índice maximiza la similitud entre dos ventanas.

Como fue explicado en la sección 2.4.1, el objetivo de los algoritmos estéreo por correlación basada en áreas, es estimar la disparidad de un punto en la imagen de referencia al comparar la ventana con dimensiones *nxm* dentro de un rango de ventanas de la imagen de búsqueda, a este rango se

le conoce como *rango de disparidad*. Se define la literal *l* como el tamaño del rango de disparidad.

La definición de los valores para *m, n, l* juega un papel muy importante para obtener mapas de disparidad que representen de la mejor manera la estructura tridimensional de la escena capturada por el sistema estéreo. Sin embargo, no existe una regla para definir dichos parámetros. En la literatura [30] se puede encontrar, (cita textual): *"Si la ventana de correlación es demasiada pequeña y no cubre suficientes variaciones en intensidad, ésta proporciona una estimación muy pobre en el mapa de disparidad, porque la razón señal a ruido SNR es baja. Si, por otro lado, la ventana es muy grande y cubre una región de la escena en la cual la profundidad varía, entonces la posición de máxima correlación o de mínima SAD (Sum of Absolute Differences) podría no proporcionar una comparación correcta debido a las diferentes transformaciones de perspectiva presentes tanto en la imagen derecha como en la imagen izquierda"*. Por otro lado, es común el uso de ventanas de tamaño no mayor a 7x7 píxeles dentro de sistemas estéreo de tiempo real [8].

El tamaño del rango de disparidad *l* tampoco debe tomarse a la ligera, *l* debe ser definido en base a la distancia entre los centros ópticos de los sensores de captura (línea base), ya que si el tamaño es definido demasiado pequeño, la estimación de la disparidad mínima no será correcta, pues la ventana de referencia nunca se compararía con la ventana de búsqueda correcta. Por otro lado, si el rango es demasiado grande, es *posible* que la estimación de la disparidad encontrada sea incorrecta debido a que la ventana de referencia se compare con la ventana de búsqueda que maximice la medida de similitud y ésta no sea la correcta. En [8], se concluye que, en la mayoría de los sistemas estéreo, el uso del rango de disparidad con valor 15 es preferido, debido al balance *rendimiento vs calidad*. También, se observa en

productos comerciales [27,19] que el uso frecuente de la línea base es del orden de 12cm. de distancia.

Por lo anterior, se define el uso de ventanas de tamaño 7x7 en la arquitectura presentada, mientras que el rango de disparidad se define de 15 píxeles. Obsérvese en la discusión más adelante, que el rango de disparidad de la arquitectura es fácilmente escalable, por lo que la distancia de la línea base puede también ser configurada de acuerdo a las especificaciones de la aplicación.

Ahora, se resumen los pasos necesarios para generar un mapa de disparidad, dado un par de imágenes estéreo rectificadas.

1. Definir los parámetros de búsqueda (*m = 7, n = 7, l = 15)*
2. Aplicar la medida de similitud SAD usando la ventana de referencia comparándola con todas las (9) ventanas de búsqueda del rango de disparidad. Son 9 ventanas de búsqueda, pues con 9 ventanas se cubren 15 píxeles.
3. Seleccionar el índice donde se encuentra el valor mínimo SAD (de los 9 disponibles). Este índice maximiza la similitud entre las ventanas comparadas. El índice del valor mínimo representa la disparidad entre las ventanas comparadas.
4. Guardar el valor generado en el paso 3 en la localidad correspondiente dentro del mapa de disparidades.
5. Repetir los pasos 2, 3 y 4 tantas veces como el tamaño de la imagen (640x480 veces).

Una vez resumidos los pasos para la aplicación del algoritmo estéreo, se discute el funcionamiento del módulo *Correlación Estéreo*.

El flujo de datos y procesamiento, debe iniciar al leer la ventana de referencia y la ventana de búsqueda, para después realizar la resta absoluta de cada píxel en la ventana de referencia contra su correspondiente en la ventana de búsqueda. Generada la ventana de diferencias absolutas (*AD*), se procede a sumar todos sus elementos obteniendo el valor “Suma de diferencias absolutas” *SAD*. Es necesario guardar el valor SAD de todas las comparaciones, para luego seleccionar el índice donde el valor SAD sea el mínimo.

El módulo *Correlación Estéreo* explota la funcionalidad del módulo *Ruteador* al hacer uso de su característica de almacenamiento. Se diseñó el módulo *Correlación Estéreo* haciendo uso de técnicas de paralelismo y etapas de *pipeline* para acelerar el algoritmo estéreo.

El módulo *Correlación Estéreo* solicita al ruteador, en lugar de píxel por píxel, una columna completa de la ventana de referencia y una de la ventana de búsqueda. El módulo *Ruteador* lee de la memoria RAM correspondiente la columna solicitada y la almacena en registros internos. Debido a las características propias de los registros internos, es posible para el módulo *Correlación Estéreo* realizar la diferencia absoluta de todos los píxeles de esas columnas en un ciclo de reloj. Dicho valor, es guardado en un bloque de memoria interno para después pasar a un componente acumulador. Por lo tanto después de leer las columnas de las ventanas, mas un ciclo para realizar la diferencia absoluta, se obtiene la SAD de las ventanas procesadas. Al final, cuando ya se generaron todos los valores SAD del rango de disparidad, los cuales son guardados en otro bloque de memoria interno (*winSAD*), se procede a seleccionar el índice donde ocurre el valor mínimo SAD. El procesamiento anterior, que genera un vector con los valores SAD (w*inSAD,*) toma lugar dentro de un componente interno del módulo *Correlación Estéreo*, llamaremos a este componente *Elemento Procesador (**PE)***.

El diagrama a bloques del *elemento procesador* se muestra en la figura 3.4.

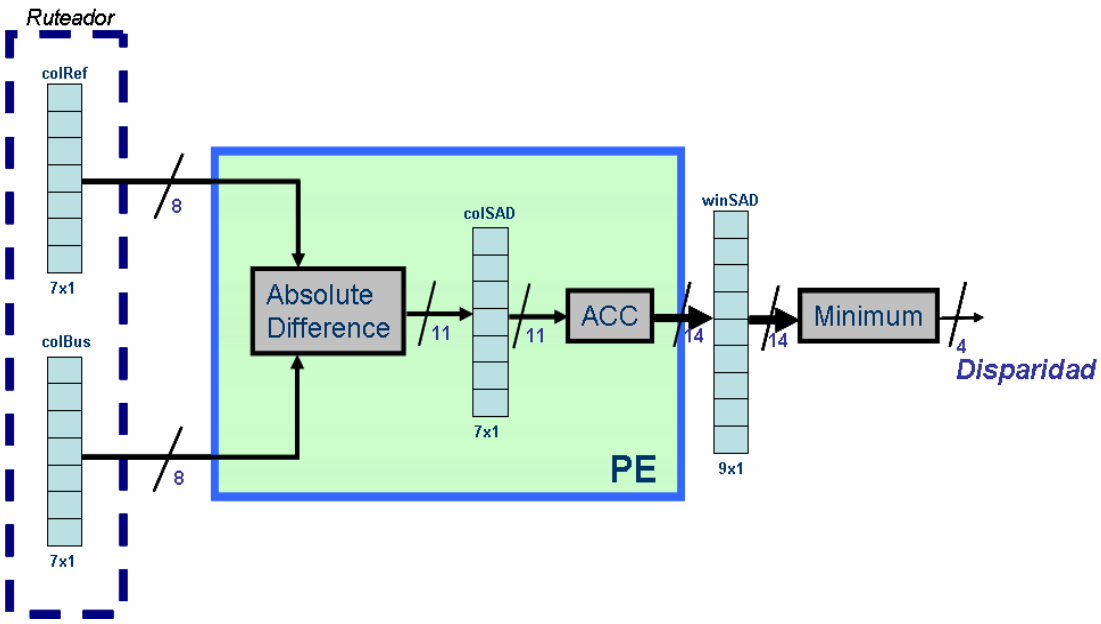

Figura 3.4 Diagrama a bloques: Elemento Procesador.

Del diseño de *PE*, y del hecho que la segunda columna de la primera ventana de búsqueda, es la primera columna para la segunda ventana de búsqueda, y así consecutivamente, surge la segunda idea para mejorar el rendimiento de la arquitectura. La idea es, si en lugar de leer una columna de la ventana de búsqueda, se leyeran dos, es posible implementar un segundo elemento procesador que ahora compare la ventana de referencia contra la segunda ventana de búsqueda, por lo que al final del procesamiento del PE, se obtendrían dos valores SAD. Esta idea da a lugar el uso de 9 PE, cada uno con su respectiva primer columna de la ventana de búsqueda, y por lo tanto al finalizar el procesamiento de cada PE, se generan los 9 valores SAD de manera paralela que se almacenan en el vector *winSAD*. Al conjunto de 9 PE lo llamaremos *Procesador Fila,* ***(RowProcessor)***.

El diagrama a bloques del procesador fila se muestra en la figura 3.5

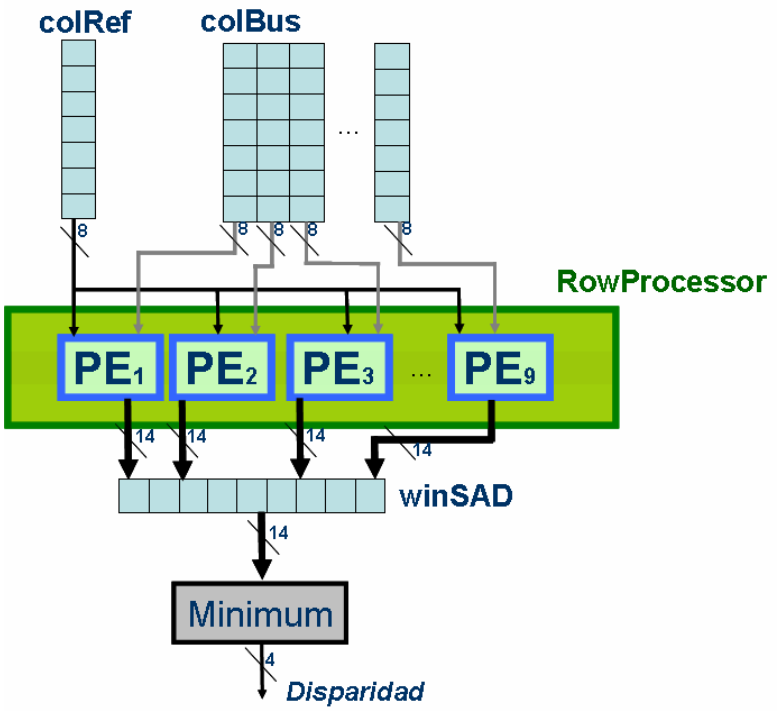

Figura 3.5 Diagrama a bloques del RowProcessor. Conjunto de 9 PE para generar 9 valores SAD en forma paralela.

Anteriormente, se mencionaba la *facilidad* con que se podría escalar la arquitectura propuesta para trabajar rangos de disparidad mayores a 15 píxeles. La arquitectura estéreo propuesta es escalable ya que solo se requiere agregar tantos elementos procesadores como se requieran para trabajar con rangos de disparidad mayores a 15 píxeles.

Así como se generó la idea anterior de los PE's, se crea una nueva idea con la que se sigue explotando las características de paralelismo del algoritmo estéreo (en realidad, la mayoría de los algoritmos de visión tienen estas características debido al procesamiento matricial). Supóngase ahora, que en lugar de leer una columna de las ventanas a procesar (es decir de tamaño 7x1), se leyera un píxel mas para cada columna (ahora sería de 8x1). Con esto, se lograría obtener cualquier columna de la segunda fila de las

imágenes estéreo. Por lo tanto es posible colocar otro procesador fila, logrando que en el tiempo que se tarda un elemento procesador en terminar su tarea, se generen dos vectores *winSAD*. Se sigue, que se pueden implementar tantos procesadores fila como se requieran, teniendo como cota superior el tamaño en altura de las imágenes estéreo.

El diagrama a bloques que representa al conjunto de procesadores fila, se muestra en la figura 3.6

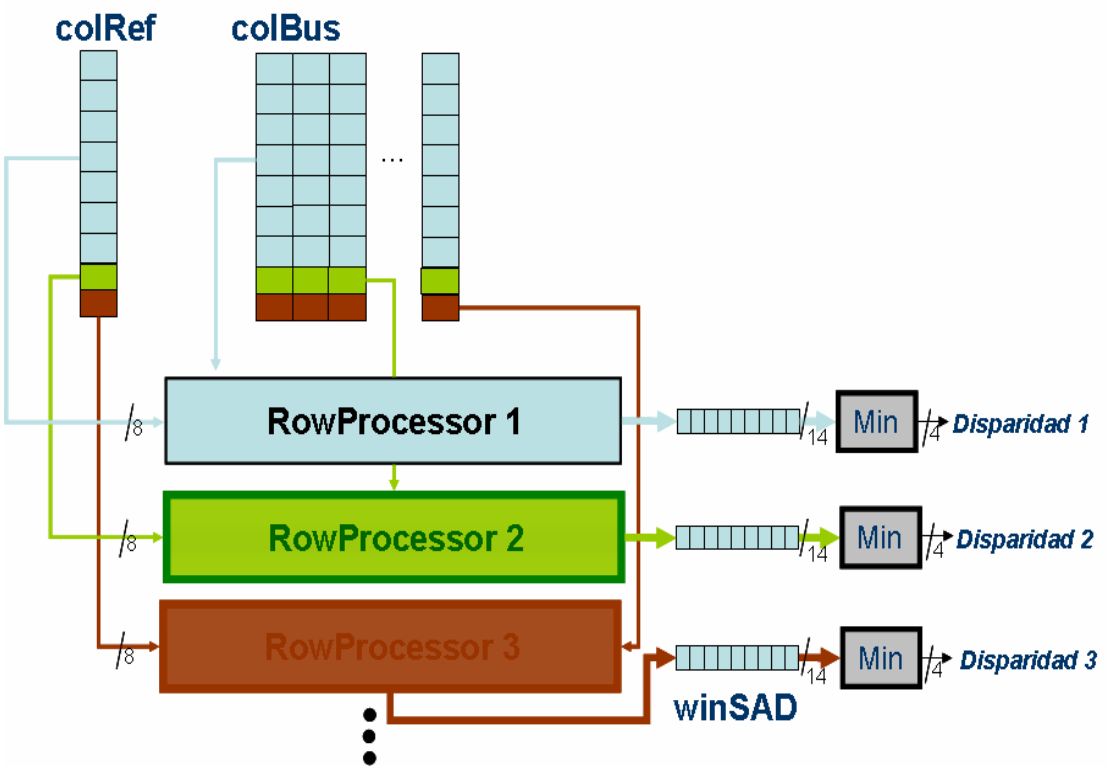

Figura 3.6 Diagrama a bloques de un conjunto de rowProcessor's.

La arquitectura estéreo es escalable tanto como la aplicación final lo requiera en términos de rendimiento y uso de recursos dentro del FPGA. Debido a esto, es necesario realizar un análisis de rendimiento, para poder determinar cuantos elementos procesadores así como procesadores fila se im-

plementarán en la etapa final. El análisis de rendimiento se muestra en la sección 3.5.

El diagrama de la figura 3.7 muestra la vista de mayor nivel del módulo *Correlación Estéreo*.

Figura 3.7 Diagrama a bloques del módulo correlación estéreo

Hasta ahora se ha mostrado el funcionamiento del módulo *Correlación Estéreo*, sin embargo, su funcionalidad reside en la capacidad de acceder a los datos correctos en los tiempos correctos. El acceso a dichos valores es una de las funciones del módulo *Ruteador*, mientras que la sincronía de todo el sistema es el trabajo del módulo *Control*. Las siguientes secciones definen el funcionamiento de los módulos *Ruteador* y *Control*.

Módulos Ruteadores para Correlación Estéreo.

El objetivo de los ruteadores es proveer los datos necesarios por los módulos de procesamiento para su funcionamiento. Para el módulo *Correlación Estéreo*, el ruteador debe esperar por la señal de pedido (request) proveniente del módulo *Correlación Estéreo* para empezar a leer los píxeles de las memorias correspondientes. Al ruteador se le especifica las coordenadas (es decir, las direcciones en la memoria) en donde debe empezar a leer las columnas, para después almacenarlas localmente en un arreglo de 7 filas por 9 columnas para la implementación de 1 procesador fila con 9 elementos

procesador. Si se implementaran 2 procesadores fila, se requeriría almacenar 8 filas por 9 columnas y así sucesivamente.

Nótese que no es necesario guardar las 15 columnas del rango de disparidad ya que cuando se necesite la décima columna (y las sucesivas), el módulo *Ruteador* está diseñado para almacenar las columnas como los registros circulares, es decir, la décima columna se almacena en la primera posición, la décimo primer columna en la segunda posición y así para el resto de las columnas. Esta técnica reduce el uso de recursos del FPGA.

El proceso del módulo *Ruteador* finaliza cuando termina de cargar las columnas requeridas, y cuando esto sucede, envía una señal de finalizado al módulo *Correlación Estéreo*. Esta señal está interconectada a la entrada de habilitación EN logrando con esto, que los módulos estén sincronizados entre ellos.

El diagrama de bloques del módulo *Ruteador* para la correlación estéreo se muestra en la figura 3.8.

Figura 3.8 Diagrama de bloques del módulo ruteador para correlación estéreo.

3.3.3 Módulo Control.

El módulo *Control* es el encargado de mantener la sincronía entre los módulos de procesamiento, así como de configurarlos. El módulo *Control*

implementa una máquina de estados que actualiza principalmente los índices para los accesos a las memorias, que serán usados por módulos *Ruteadores*, así como habilitar o deshabilitar los demás módulos para llevar el flujo de información.

Por lo tanto, el módulo *Control*, tiene como única entrada a la señal de reloj de la arquitectura. La funcionalidad de este módulo es realizar el conteo de ciclos necesarios para acceder a las memorias, tanto para lectura de los píxeles como para escritura de los resultados arrojados por los módulos de procesamiento. El módulo *Control* tiene como señales de salida cuatro buses de control, uno para el módulo *Transformación Geométrica*, otro para el módulo *Correlación Estéreo*, y dos más para los módulos *Ruteadores*. Obsérvese en la figura 3.1, que el módulo *Control* deberá ser actualizado con el fin de sincronizar los módulos periféricos agregados a la arquitectura, esto es, manejar los controladores de los sensores así como el controlador del dispositivo USB.

Los buses de control para cada módulo *Ruteador* contiene la información de las direcciones para acceder a las memorias, así como la señal de escritura o lectura de las mismas. También contienen las señales para habilitar o deshabilitar los módulos *Ruteadores.* Por otro lado, los buses de control para los módulos de procesamiento, están formados por señales de habilitación así como también de contadores para el módulo de *Correlación Estéreo* que permiten controlar el flujo de datos para el procesamiento de las comparaciones de las ventanas creadas. El diagrama de bloques del Módulo *Control* se muestra en la figura 3.9.

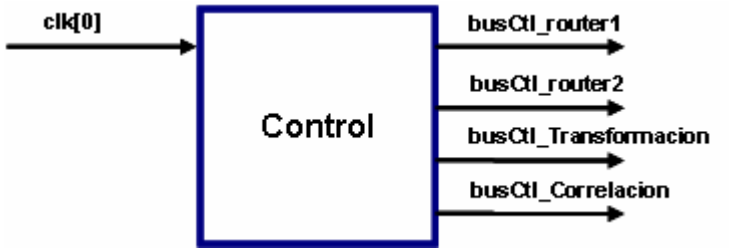

Figura 3.9 Diagrama de bloques: Módulo Control.

3.4 Extensiones propuestas a la arquitectura estéreo.

Durante esta sección se revisarán tres extensiones propuestas a la arquitectura estéreo. Estas extensiones tienen como objetivo mejorar el rendimiento y calidad de las estimaciones de los índices de disparidad generados por la arquitectura. Una extensión se propone para el módulo *Transformación Geométrica* y dos más para el módulo *Correlación Estéreo.*

3.4.1 Extensión para el módulo *Transformación Geométrica*.

Recordando la ecuación 3.1, se identifica que el valor de las coordenadas correctas dependen del valor de un polinomio de grado 6 con 10 variables. Las variables del polinomio son lo parámetros de la cámara, dichos parámetros describen la geometría interna de los sensores, así como la geometría existente entre los sensores con los objetos en el espacio tridimensional.

Los parámetros de la cámara geométrica se obtienen a través del método de calibración (véase el capítulo 2.3.6). Los resultados del método de calibración usando cualquiera de los diferentes modelos matemáticos para

resolverlo, arrojan valores decimales, es decir, los parámetros de la cámara se definen en el espacio de los números reales.

De lo expuesto en el párrafo anterior, se establece que las coordenadas calculadas por el modelo matemático descrito en la ecuación 3.1, se encuentran representadas por números reales. Este hecho representa un problema, debido que el espacio de las coordenadas en las imágenes digitales están codificadas con número enteros sin signo. La figura 3.11 representa al problema descrito de manera gráfica.

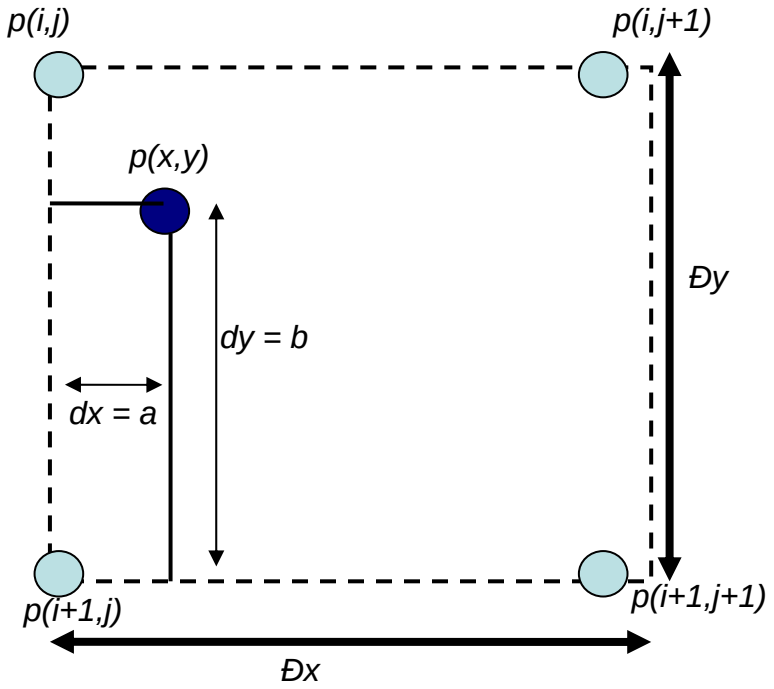

Figura 3.11 Diagrama de las coordenadas *(x,y)* en punto fijo. El píxel *p* tiene coordenadas de tipo real *x,y*. Es necesario ajustar el píxel *p* a las coordenadas *i,j* de tipo entero. *dx* y *dy* es la distancia real del píxel *p* a las posiciones *i,j*.

Existen diferentes maneras de atacar el problema descrito en la figura 3.11. La técnica comúnmente usada por su sencillez computacional, es truncar el valor de las coordenadas calculadas. Sin embargo, el truncamiento de

valores reales, siempre arrastra un error, en éste caso, el error se observa en la calidad de las imágenes transformadas. Cuando se realiza un truncamiento en las coordenadas calculadas, se observa el efecto de escalera, es decir, si la imagen distorsionada cuenta con líneas rectas, éstas al ser corregidas no mantendrán su aspecto recto, sino que se observarán como una escalera, donde la amplitud de sus escalones está definida por la estimación correcta de las coordenadas.

Otra técnica usada es seleccionar el píxel más cercano a *p(x,y)*, usando como métrica de distancia la representada por la distancia euclidiana. El efecto escalera se sigue observando, aunque más suavizado. En este trabajo se propone la extensión al módulo *Transformación* Geométrica para realizar una interpolación bilineal de los píxeles vecinos a *p(x,y)*. Dado que el píxel *p(x,y)* contiene un valor de intensidad influido por sus vecinos, es necesario fusionar de alguna manera los valores de intensidad de los píxeles vecinos. Una manera es al tomar el promedio de los valores de intensidad de los píxeles vecinos, sin embargo, realizar un promedio ponderado (asignar pesos a los vecinos de acuerdo a la cercanía del píxel original), nos permite obtener mejores resultados. Este promedio ponderado se le conoce como interpolación bilineal subpíxel.

La interpolación bilineal obtiene mejores resultados pero también tiene mayor costo computacional. Esta técnica asigna al píxel en cuestión un valor medio ponderado de las intensidades de los cuatro píxeles vecinos. Los pesos se asignan de acuerdo a las distancias existentes y se calculan a través de las siguientes ecuaciones:

$$a_1 = \left(1 - \frac{dx}{Dx}\right)\left(1 - \frac{dy}{Dy}\right) \qquad a_2 = \left(\frac{dx}{Dx}\right)\left(1 - \frac{dy}{Dy}\right) \tag{3.3}$$

$$a_3 = \left(1 - \frac{dx}{Dx}\right)\left(\frac{dy}{Dy}\right) \qquad a_4 = \left(\frac{dx}{Dx}\right)\left(\frac{dy}{Dy}\right)$$

Dado que Dx = Dy = 1, las ecuaciones 3.3 se simplifican a:

$$a_1 = (1 - dx)(1 - dy) \qquad a_2 = dx(1 - dy) \qquad (3.4)$$
$$a_3 = (1 - dx)(dy) \qquad a_4 = dxdy$$

Por lo tanto, la función que estima el valor del píxel se calcula de la siguiente manera:

$$p(x, y) = a_1 p(i, j) + a_2 p(i, j+1) + a_3 p(i+1, j) + a_4 p(i+1, j+1) \qquad (3.5)$$

3.4.2 Extensiones para el módulo *Correlación Estéreo*.

Interpolación

De la descripción de la arquitectura estéreo, en particular del módulo *Correlación Estéreo*, se define que el rango de disparidad es de 15 píxeles, por lo tanto, existen 15 índices, que, de entre ellos, se selecciona aquel que contenga el valor mínimo SAD que maximiza la similitud entre esas dos ventanas. De lo anterior se deduce, que la arquitectura estéreo propuesta, genera mapas de disparidad con valores de profundidad de 15 pasos, por lo que cada valor en el eje Z, se codifica con 4 bits.

Sin embargo, la resolución del mapa de disparidad puede ser mejorada usando técnicas de interpolación. De manera similar a la técnica discutida en la sección anterior, se puede implementar la técnica de interpolación para mejorar la calidad de resultados, en este caso, la interpolación se deberá

aplicar a la selección del índice del valor SAD mínimo. Ya calculados los valores SAD de todo el rango de disparidad, se puede estimar una función que pase por la curva denotada por el vector de los valores SAD (en este caso, el vector *winSAD*). Usando esta función, se pueden calcular valores SAD intermedios, con los cuales, se puede establecer un número mayor de índices de disparidad. Encontrar la función que describa la curva del vector SAD, así como calcular valores intermedios es una técnica de interpolación. Por lo tanto, se propone una extensión al módulo *Correlación Estéreo* para mejorar la resolución del mapa de disparidad, es decir, para generar *mapas de disparidad con resolución subpíxel.*

Correlación con imágenes de color.

En [8] se realizó una investigación que demuestra que el uso de imágenes estéreo codificadas en el espacio de color, permite generar mejores mapas de disparidad. Cabe destacar que en la investigación de las arquitecturas hardware para el procesamiento estéreo, no se ha encontrado una arquitectura que trabaje con imágenes codificadas en espacios de color.

En [8] se propone un protocolo para evaluar el uso de imágenes de color para la correlación estéreo. Dicho protocolo define tres métodos diferentes para realizar la correlación usando imágenes de color. También se definen nueve espacios de color para el uso de los métodos anteriores. Los nueve espacios de color usados en los métodos son:

1. RGB
2. $L^*u^*v^*$
3. $L^*a^*b^*$
4. AC_1C_2
5. YC_1C_2

6. HSI
7. $I_1I_2I_3$
8. $H_1H_2H_3$
9. XYZ

Los tres métodos propuestos para realizar la correlación estéreo sobre imágenes a color se resumen a continuación.

Método 1.

Realizar la correlación estéreo sobre cada canal del espacio de color de las imágenes estéreo, para después fusionar los tres mapas de disparidad generados, usando el modelo matemático: *Fusión de Belli* [5].

Método 2.

Realizar el análisis del componente principal *PCA* sobre las imágenes estéreo, para después aplicar la correlación estéreo sobre el primer componente principal.

Método 3.

Realizar la correlación usando directamente las imágenes de color. En éste método, se proponen las reglas para adaptar las diferentes medidas de similitud.

El autor de la investigación [8] concluye su trabajo mostrando resultados estadísticos de las diversas pruebas realizadas a los tres métodos propuestos. Los resultados son:

"Se puede concluir que el método 1 con el espacio de color XYZ o el método 3 con el espacio de color $H_1H_2H_3$ son generalmente los mejores (64% de los casos)".

Dada la lectura anterior, se propone una extensión a la arquitectura estéreo. A continuación se realiza una discusión para la implementación de los dos métodos propuestos en [8].

La implementación en hardware del método 1, teniendo como base la arquitectura estéreo presentada en esta investigación requiere replicar el módulo *Correlación Estéreo* y su módulo *Ruteador* tres veces, con el fin de entregar 3 mapas de disparidad usando los canales de cada imagen. Será necesario diseñar un módulo nuevo que permita realizar la fusión de los tres mapas de disparidad. Por último, es necesario actualizar la función del módulo *Control*, dado que el actual funcionamiento se realiza sobre memorias de 32 bits que almacenan 4 píxeles de 8 bits cada uno. Por lo tanto el control se debe modificar, para leer un solo píxel (de 24 bits) de la memoria.

Para que el método 1 trabaje con el espacio de colores XYZ, es necesario transformar los píxeles de entrada, los cuales, no necesariamente están codificados en XYZ. La transformación de espacios de color es relativamente sencilla, debido a que se basa en la multiplicación matricial. Se muestra la función matricial necesaria para transformar del espacio de color RGB (espacio de color comúnmente usado en el procesamiento de imágenes) al espacio XYZ:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.3}$$

La implementación en hardware de la transformación matricial descrita en la ecuación 3.3, a pesar de la simplicidad de la transformación, no es trivial en hardware, debido al uso de números de punto fijo. Es necesario, por lo tanto, un análisis de rendimiento antes de optar por la implementación del método 1 en hardware.

Por otro lado, la implementación en hardware del método 3, usando como base la arquitectura estéreo presentada en esta investigación, requiere de la modificación al módulo *Correlación Estéreo*. Las modificaciones necesarias son, en lugar de realizar las operaciones matemáticas: resta y suma para píxeles de 8 bits, implementar las adaptaciones mostradas en [8] para las operaciones matemáticas previamente mencionadas. De la misma manera que se necesita la modificación de control para el método 1, también se requiere actualizar la función del módulo *Control*, para leer un solo píxel por localidad de la memoria en lugar de 4 píxeles por localidad.

La transformación del espacio de color para el método 3 también es una multiplicación matricial. Se muestra la ecuación para la transformación del espacio de color RGB al espacio de color $H_1H_2H_3$.

$$\begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.4}$$

A diferencia de la transformación expresada en la ecuación 3.3, la implementación de esta ecuación (3.4) en hardware resulta mucho más sencilla, dado que sólo se requiere realizar sumas y restas sobre los píxeles de entrada, así como realizar un desplazamiento de 1 bit a la derecha para realizar la división entre 2. Se requiere de un análisis más profundo para selec-

cionar el método más factible para su implementación en hardware, pero el análisis realizado en esta sección, apunta que el método 3 con el espacio de color $H_1H_2H_3$ es el más indicado para la implementación en hardware.

La figura 3.12 muestra el diseño de la arquitectura estéreo con las adaptaciones propuestas arrojadas a partir de la discusión de la sección 3.4.2. Los módulos sombreados son a los que será necesario modificar o crear, para obtener una arquitectura estéreo que implementa la correlación usando imágenes a color. La implementación de las modificaciones descritas en esta extensión se propone como trabajo a futuro en el capítulo 5.

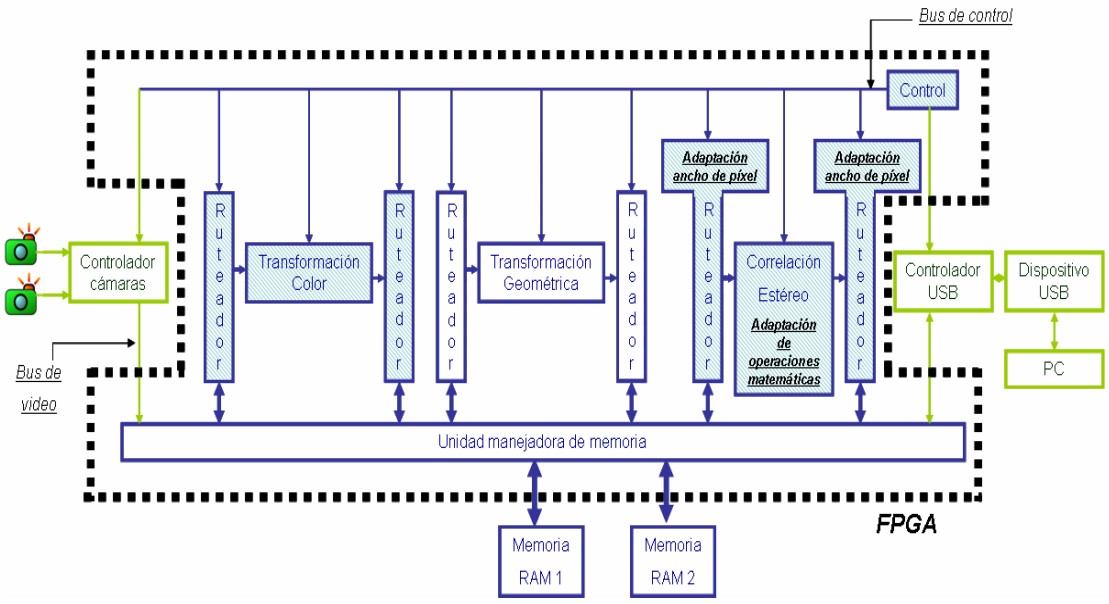

Figura 3.12 Arquitectura estéreo usando imágenes de color.

3.5 Análisis de rendimiento.

Esta sección está dedicada a realizar un análisis de rendimiento de la arquitectura estéreo. El objetivo del análisis de rendimiento es calcular cuan-

tos ciclos de reloj tarda la arquitectura en generar un mapa disparidad dado un par de imágenes estéreo no rectificadas.

Para realizar el análisis de rendimiento, es necesario detectar qué módulo o conjunto de módulos tardan más ciclos en generar su resultado. Debido a que los módulos de procesamiento de la arquitectura trabajan en paralelo, y controlados para realizar un flujo de datos usando la técnica pipeline; se debe identificar, cuál de estos módulos requiere más ciclos de reloj para generar su resultado.

Como ya ha sido mencionado, el módulo *Transformación Geométrica* junto con su *Ruteador*, realiza un mapeo de direcciones. Es decir, a la entrada de un píxel a la arquitectura, el módulo *Transformación Geométrica* y su *Ruteador* toman un ciclo para leer el píxel actual, en el siguiente ciclo lee la dirección correcta y en el último ciclo reescribe el píxel leído en su posición correcta. Por lo tanto, el módulo *Transformación Geométrica* requiere de 3 ciclos para escribir un píxel en su posición correcta.

El módulo *Correlación Estéreo* y su *Ruteador* requieren de un análisis más profundo y como se verá más adelante, estos módulos consumen mayor número de ciclos para realizar su tarea, por lo tanto, son estos módulos los que determinarán el número de ciclos necesarios para obtener un mapa de disparidad.

Se analiza en primera instancia, el componente básico del módulo *Correlación Estéreo*, el elemento procesador. El PE requiere para iniciar su procesamiento, tener disponible el conjunto de registros que almacenan alguna columna de las ventanas procesadas. El ruteador de la correlación estéreo es el encargado de generar dichos conjuntos. Debido a que la memoria almacena 4 píxeles por localidad, el ruteador necesita de 2 ciclos para leer los

7 píxeles de la ventana a procesar (en realidad, dado que en un ciclo se accede a 4 píxeles, en 2 ciclos se accede a 8 píxeles). Adelantando información descrita en el capítulo 4, en donde se establece que la arquitectura estéreo implementa 3 procesadores fila, por lo tanto es necesario contar con columnas de 9 localidades para procesar 3 ventanas. De lo anterior, se actualiza la información de rendimiento para el ruteador, es decir, ahora el ruteador necesita 3 ciclos de reloj para acceder a 9 píxeles.

En la siguiente etapa del pipeline, se inicia el procesamiento de PE. El PE realiza la diferencia absoluta de los 7 píxeles correspondientes en un solo ciclo debido a que el acceso a los registros se puede realizar de manera paralela. En el segundo ciclo de procesamiento, el PE realiza la suma del vector que contiene la columna de diferencias absolutas, y acumula el resultado en el registro correspondiente. El PE tiene registros internos de control que necesitan ser actualizados, estos registros ocupan el tercer ciclo de reloj del procesamiento. Obsérvese que después de 3 ciclos se obtiene la SAD de un par de columnas de las ventanas procesadas. Este procedimiento se debe repetir tantas veces como columnas existan en las ventanas a correlacionar. Dado que la arquitectura realiza la correlación sobre ventanas de tamaño 7x7, se concluye que el componente requiere de 21 ciclos para generar un valor SAD de dos ventanas procesadas. La tercera y última etapa del pipeline, se encarga de encontrar el índice (disparidad) donde ocurre el valor mínimo, este proceso requiere de 3 ciclos. Debido a que el estado de pipeline más grande es de 21 ciclos, el análisis de rendimiento para el elemento procesador entrega como resultado que: ***Para generar un índice de disparidad se requieren de 21 ciclos***.

En general, se puede definir una ecuación para determinar el número de ciclos necesarios para generar el índice de disparidad, la ecuación es:

$$\#ciclos = max(etapa1, etapa2, etapa3) \quad (3.5)$$

donde

$$etapa1 = ceil\ (n/\ 4\ ) \quad (3.6)$$

$$etapa2 = 3\ x\ m \quad (3.7)$$

$$etapa3 = ceil(\ l/\ 3\ ) \quad (3.8)$$

n = altura de la ventana de correlación
m = ancho de la ventana de correlación
l = longitud del vector winSAD

Observando la figura 3.6, se deduce que la arquitectura estéreo entrega tantos índices de disparidad como procesadores fila se implementen. Es decir, implementando *k* procesadores fila se generarán *k* índices de disparidad de manera paralela, por lo tanto, ***la arquitectura estéreo genera k índices de disparidad en 3 x m ciclos.***

Se concluye, a partir del análisis de rendimiento, que para procesar imágenes estéreo no rectificadas, de resolución VGA, ***la arquitectura estéreo requiere de (640x480x3xm) / k ciclos para entregar un mapa de disparidad de resolución VGA.***

El resultado anterior, se puede generalizar en una ecuación que calcule los ciclos necesarios para entregar un mapa de disparidad de un par de imágenes estéreo no rectificadas de tamaño *MxN*, usando *k* procesadores fila. La ecuación es:

$$\#ciclos = (MxNx3xm) / k \quad (3.9)$$

La función descrita en la ecuación 3.9 establece que a mayor número de procesadores fila *k*, menor será el número de ciclos necesarios para procesar un par de imágenes estéreo. La figura 3.13 grafica la ecuación 3.9, con parámetros M = 640, N = 480, y m = 7.

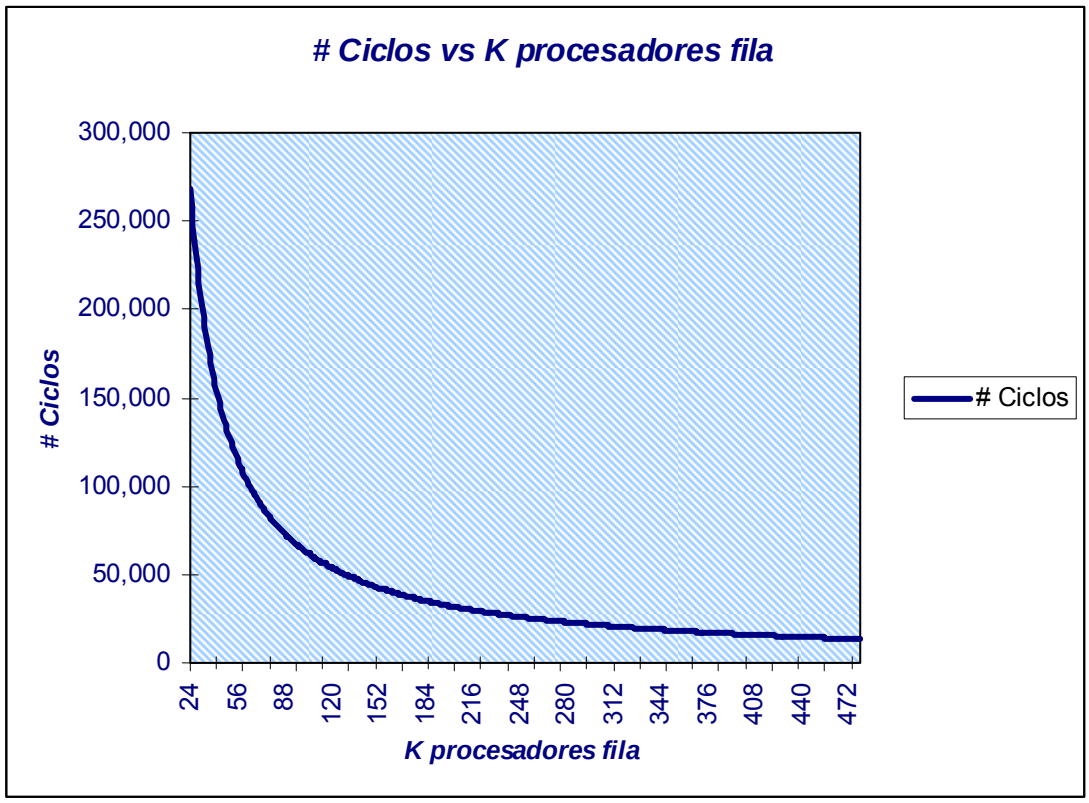

Figura 3.13 # Ciclos vs Procesadores fila.

La selección de *k* se debe establecer analizando el compromiso entre la velocidad de procesamiento y la ocupación de la arquitectura con *k* procesadores fila dentro del FPGA.

3.6 Discusión.

Durante el capítulo 3, se ha descrito la arquitectura hardware para la recuperación 3D. Se presentó el diseño global de la arquitectura estéreo, la cual consiste principalmente, en dos módulos de procesamiento: *Transformación Geométrica* y *Correlación Estéreo*.

El módulo *Transformación Geométrica* realiza el procesamiento necesario para generar un par de imágenes estéreo rectificadas a través de tablas de mapeo.

El módulo *Correlación Estéreo* resuelve el problema de correspondencia estéreo al usar el método local de correlación basada en área. El funcionamiento del módulo de correlación se basa en el procesamiento de los elementos procesador. El número *k* de conjuntos de elementos procesador definen el número de ciclos necesarios para procesar un par de imágenes estéreo. Mientras más procesadores se implementen, menor será el tiempo necesario para procesar el par estéreo. La selección de *k* no es trivial, y se debe establecer un compromiso entre velocidad de procesamiento y ocupación de lógica en un dispositivo FPGA.

La sección 3.4 discute y plantea extensiones a los algoritmos analizados en este trabajo de investigación. La implementación en software de las extensiones discutidas muestra que los resultados arrojados por el diseño de la arquitectura hardware propuesta en este trabajo pueden ser mejorados. Un análisis preliminar permite establecer las modificaciones necesarias al diseño propuesto con el fin de implementar las extensiones de la sección 3.4.

El siguiente capítulo 4, describe la implementación de la arquitectura estéreo en base al diseño del capítulo 3. Los resultados de síntesis de alto

nivel, permiten establecer un compromiso entre velocidad de procesamiento y ocupación en el FPGA con lo que es posible determinar el número *k* de procesadores fila. Se muestran también los resultados de procesamiento de la arquitectura hardware a nivel simulación, así como comparativos con los resultados obtenidos por la implementación software.

El capítulo 4 concluye mostrando los resultados preliminares de las extensiones propuestas en la sección 3.4 implementadas en software.

Capítulo 4

Implementación y Resultados.

4.1 Introducción.

El capítulo 4 está dedicado a presentar los resultados obtenidos durante toda la investigación de este trabajo. El capítulo 4 está dividido de la siguiente manera: en la sección 4.2 se describe la metodología de prueba llevada a cabo con el fin de validar los requerimientos impuestos en el capítulo 3, también se describen los pasos realizados para comprobar el funcionamiento de las extensiones propuesta en el documento.

La sección 4.3 muestra los resultados obtenidos por las implementaciones en software y por las simulaciones de la arquitectura hardware. Resultados preliminares de las implementaciones en software de las extensiones descritas en la sección 3.4 también son mostrados en la sección 4.3.

En la sección 4.4 se muestran los resultados de síntesis de alto nivel arrojados por la herramienta DK4 de Celoxica. Los resultados de síntesis permiten establecer el número de procesadores fila que se requiere implementar para alcanzar el balance del compromiso entre velocidad de procesamiento y recursos utilizados del dispositivo FPGA.

Se finaliza el capítulo 4 con una discusión derivada de los datos mostrados en el capítulo 4. En esta discusión se plantea la posibilidad de implementar la arquitectura estéreo en un dispositivo FPGA con mayor número de recursos para una aplicación con imágenes mayores a un megapíxel.

4.2 Implementación.

Las herramientas ocupadas para el desarrollo de este trabajo se enlistan a continuación:

1. Tarjeta de desarrollo ML403 de la empresa Xilinx, que contiene:
    a. Virtex 4 modelo XC4VFX12.
    b. DDR SDRAM de 64MB con bus de 32 bits capaz de correr hasta 266Mhz.
    c. Cristal oscilador de 100MHz.
2. Cámara Celoxica de resolución VGA.
3. Handel-C DK4 como lenguaje de descripción de hardware.
4. Matlab 7.0.1 como herramienta software.

El dispositivo FPGA tiene como características, entre otras:

- Dispositivo con capacidad de 10,944 Flip-Flops (1,368 bloques para configuración lógica CLB's, cada CLB con 4 slices, 5,472 slices en total).
- Máxima memoria RAM distribuida 86 Kbits.
- Máxima frecuencia de reloj 100 MHz.

Para validar el funcionamiento de la arquitectura estéreo, es necesario comprobar los resultados de los módulos de procesamiento junto con sus ruteadores. Debido que no se cuenta con la interfaz necesaria que genere un par de imágenes estéreo sin rectificar, la validación del módulo *Transformación Geométrica* se realizó usando el modelo matemático expuesto por la ecuación 2.3, que describe la corrección radial. Dado que el módulo *Transformación Geométrica* realiza un mapeo de coordenadas distorsionadas a las corregidas, es condición necesaria y suficiente demostrar que el mapeo es

funcional. Por otro lado, la validación del módulo *Correlación Estéreo*, se realizó usando imágenes estéreo rectificadas sintéticas.

Para comparar las imágenes arrojadas por la arquitectura estéreo, se implementaron en software los algoritmos para la corrección radial y para la correlación estéreo (usando como medida de similitud la SAD, con el criterio de selección el mínimo, y con los parámetros de búsqueda rango de disparidad 15 píxeles y ventanas de búsqueda de tamaño 7x7). Las implementaciones en software fueron realizadas con la herramienta MATLAB. De la misma manera, se implementaron los algoritmos de interpolación bilineal para la transformación geométrica, así como la correlación usando imágenes de color basándose en los métodos descritos en la sección 3.4.2.

4.3 Resultados SW / Simulación HW.

Debido a la falta de interfaces con sensores de imágenes estéreo, los resultados que en esta sección se muestran, son los resultados generados por la arquitectura estéreo a nivel simulación. La herramienta de simulación utilizada es uno de las herramientas integradas en el paquete de desarrollo de Handel-C.

4.3.1 Resultados *Transformación Geométrica.*

Simulación HW.

Los pasos que se realizaron para llevar a cabo las pruebas al módulo *Transformación Geométrica* fueron:

1. Adquisición de imágenes usando la cámara Celoxica de resolución 640x480 píxeles.
2. Las imágenes adquiridas son enviadas a una computadora personal, usando la herramienta CapEth (Herramienta desarrollada en el laboratorio de FPGAs del Instituto Nacional de Astrofísica, Óptica y Electrónica). CapEth es un software que permite capturar una imagen a través de la cámara Celoxica para después enviarla a una computadora personal por medio de los puertos Ethernet.
3. Se generaron las tablas de mapeo con la herramienta MATLAB.
4. Se realizaron pruebas de simulación del módulo usando la herramienta Handel-C.

A continuación, se muestran los algunos de los resultados obtenidos por la simulación del módulo *Transformación Geométrica.*

Figura 4.1 Primera imagen capturada

Figura 4.2 Segunda imagen capturada

Figura 4.3 Primera imagen sin distorsión

Figura 4.4 Segunda imagen sin distorsión.

Las figuras 4.1 y 4.2 muestran imágenes capturadas con la cámara Celoxica con distorsión radial. Las figuras 4.3 y 4.4 muestran las imágenes procesadas con la corrección radial.

Interpolación SW.

En la sección 3.4.1 se plantea solucionar el problema de coordenadas representadas en el espacio de los números reales aplicando la técnica de interpolación bilineal, es decir, haciendo uso del promedio ponderado de los vecinos del píxel procesado.

Se realizó la adaptación necesaria al algoritmo de transformación geométrica implementado en software, para obtener resultados preliminares y así determinar si la extensión descrita en la sección 3.4.1 es válida para una futura modificación a la arquitectura estéreo.

De la misma manera que en las secciones anteriores, las pruebas se realizan usando imágenes sintéticas, en esta ocasión, se generó una imagen "Tablero de ajedrez". Los pasos aplicados para generar los resultados preliminares fueron los siguientes:

1. Adaptación al algoritmo de transformación geométrica. Se implementa la interpolación bilineal en software.
2. Creación de la imagen sintética "Tablero de ajedrez".
3. Se aplica distorsión radial a la imagen del punto 2.
4. Se aplica corrección radial sin interpolación bilineal a la imagen del punto 3.
5. Se aplica corrección radial con interpolación bilineal a la imagen del punto 3.

6. Se comparan los resultados del punto 4 y punto 5, al aplicar la siguiente métrica.
    a. Error promedio por píxel. Nótese que en el mejor de los casos, el error promedio por píxel es 0, mientras que en el peor de los casos es 255, para píxeles codificados en niveles de gris.

Las figuras 4.5, 4.6 y 4.7 muestran la imagen sintética de prueba, la imagen distorsionada y la imagen corregida, respectivamente.

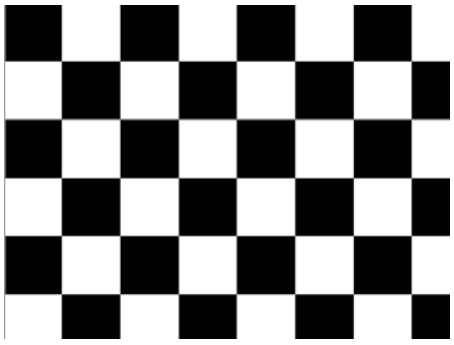

Figura 4.5 Tablero de ajedrez.

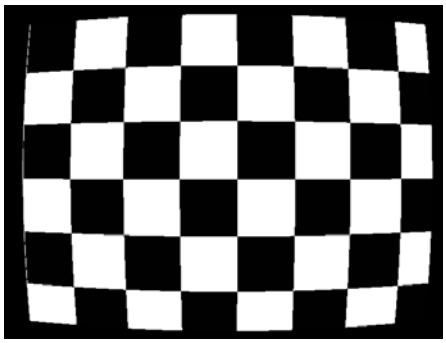

Figura 4.6 Tablero distorsionado

.

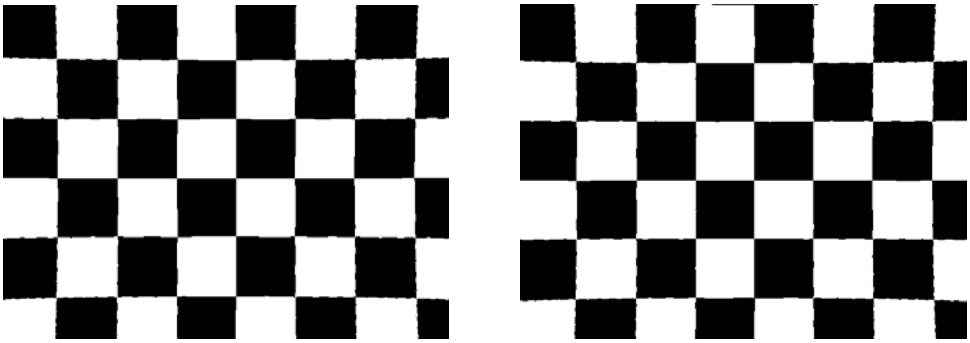

Figura 4.7 Correcciones radial. Izquierda sin interpolación, derecha con interpolación. La diferencia del error en las estimaciones es a nivel subpíxel, por lo que no fácilmente se percibe.

Para calcular las métricas de error, se realizó la diferencia absoluta de cada píxel en las imágenes corregidas menos la imagen real. Se suman todos los valores de cada localidad en la matriz resultante, y el valor calculado se divide entre el número de valores, es decir, entre 640x480. La ecuación para el error promedio por píxel MPE es la siguiente:

$$\text{MPE} = \frac{\sum_{x=0}^{639}\sum_{y=0}^{479} abs\left(I_D(x,y) - I_C(x,y)\right)}{640x480} \qquad (4.1)$$

Donde $I_D(x,y)$ y $I_C(x,y)$ son las intensidades de los píxeles de las imágenes distorsionadas y corregidas respectivamente. Los valores de las métricas son:

- MPE en la imagen sin interpolación bilineal: 6.6843 píxeles
- MPE en la imagen con interpolación bilineal: 5.474 píxeles

Los resultados de las métricas comprueban que, para este caso, la interpolación mejora la calidad de la estimación, sin embargo, es necesario un mayor análisis para corroborar la hipótesis, por ejemplo, usar imágenes con texturas, entre otras.

4.3.2 Resultados *Correlación Estéreo.*

Simulación HW.

Los pasos que se realizaron para llevar a cabo las pruebas al módulo *Correlación Estéreo* fueron:

1. Adquisición de imágenes estéreo rectificadas sintéticas.
2. Aplicación en software del algoritmo de correlación estéreo usando la medida de similitud SAD, con el criterio del mínimo, usando como parámetros, rango de disparidad 15 píxeles, y ventanas de búsqueda de tamaña 7x7.
3. Se realizaron pruebas de simulación del módulo usando como herramienta Handel-C.

Para comparar los resultados generados por la implementación en software contra los resultados de la implementación en hardware se realiza la suma de diferencias absolutas entre las dos imágenes resultantes. Los errores entre una implementación en hardware con la de una implementación en software, se debe principalmente al uso de operaciones matemáticas con números reales. La representación decimal que tienen las computadoras personales o estaciones de trabajo es fija, y a menos que la arquitectura hardware implementada alcance dicha representación decimal, siempre existirá error en los cálculos matemáticos. En el caso particular del algoritmo de correlación estéreo, no se utilizan números reales, por lo que el comportamiento de ambas implementaciones debe de ser el mismo.

Al aplicar la suma de diferencias absolutas entre los mapas de disparidad generados en software y los generados por la simulación de hardware se obtiene como valor de error 0. El valor del error calculado valida el funcionamiento de la arquitectura hardware con respecto a la implementación en software. Nótese que el valor del error calculado por la comparación entre las implementaciones de hardware y software no implica que los mapas de disparidad representen de forma exacta la estructura tridimensional. Los mapas de disparidad son *estimaciones* de la estructura tridimensional.

Nótese, que los mapas de disparidad no representan colores, si no cada píxel, representa la profundidad (inversa) que éste tiene. Para una mejor apreciación de los resultados, los mapas de disparidad densos que se muestran en esta sección, están representados en un espacio de color, de esta manera, se distinguen los diferentes niveles de profundidad encontrados.

Se muestran tres diferentes simulaciones usando 3 pares estéreo rectificados y sintéticos. Cada simulación tiene como parámetros de búsqueda, rango de disparidad 15 píxeles, ventanas de búsqueda de tamaño de 7x7 píxeles. Dado que las imágenes estéreo son sintéticas, se cuenta con los mapas de disparidad reales (Ground-truth). Se muestran los resultados de MPE para cada ejemplo comparado con el Ground-truth.

Prueba 1. Imágenes estéreo rectificadas sintéticas: *Tsukuba.*

Figura 4.8. Par estéreo sintético: Tsukuba y Ground-truth.

Resultado 1. Mapa de disparidad *Tsukuba* obtenido por la simulación HW.

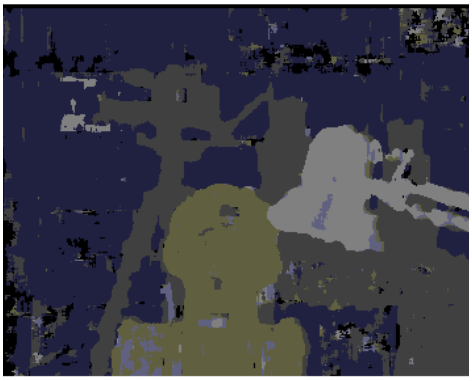

Figura 4.9 Mapa de disparidad HW: Tsukuba. **MPE = 2.4442 píxeles**

Prueba 2. Imágenes estéreo rectificadas sintéticas: *Teddy.*

Figura 4.10. Par estéreo sintético: Teddy y Ground-truth.

Resultado 2. Mapa de disparidad *Teddy* obtenido por la simulación HW.

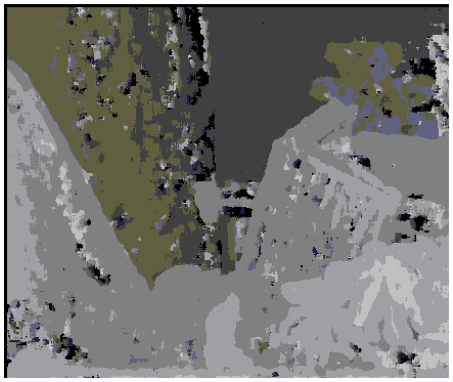

Figura 4.11 Mapa de disparidad HW: Teddy. **MPE = 2.7242 píxeles**

Prueba 3. Imágenes estéreo rectificadas sintéticas: *Map.*

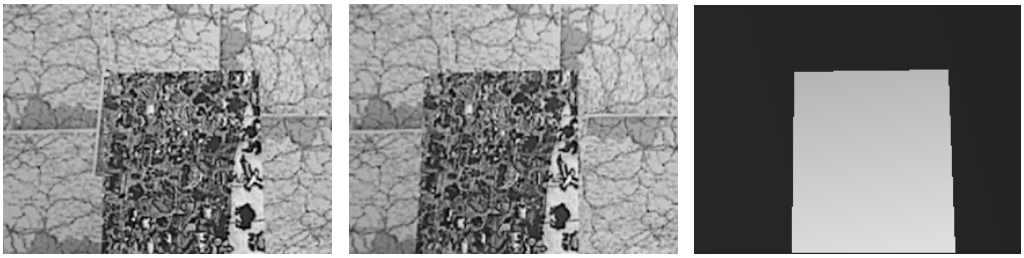

Figura 4.12. Par estéreo sintético: Map y Ground-truth.

Resultado 3. Mapa de disparidad *Map* obtenido por la simulación HW.

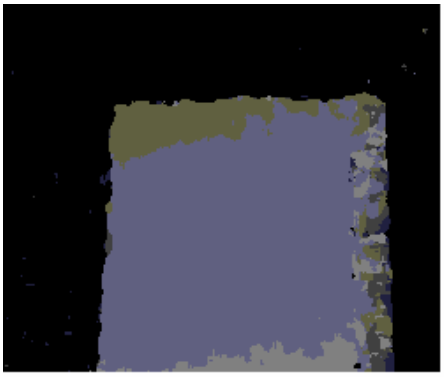

Figura 4.13 Mapa de disparidad HW: Map. **MPE = 3.3342 píxeles**

Correlación con imágenes de color. Implementación en software.

En la sección 3.4.2 se mencionó que en la literatura [8] se sugiere que el uso de imágenes codificadas en determinados espacios de color en los algoritmos de correspondencia estéreo, en específico, los basados en correlación por área, pueden generar mapas de disparidad con mejor calidad en la estimación de la profundidad.

En este trabajo, se propone como extensión a la arquitectura estéreo realizada, la adaptación del algoritmo de correlación usando imágenes estéreo con color. Más allá de la propuesta, se presentan en este documento, pruebas preliminares para determinar si los mapas de disparidad densos muestran mejores estimaciones usando imágenes de color. Para poder establecer una métrica para medir las mejoras (si las hay) en los mapas de densidad se debe de contar con métodos que permitan establecer puntos de comparación. No es alcance de esta investigación proponer dichos métodos, sin embargo, en [8] se sugieren algunos de ellos.

Esta sección muestra los resultados de las pruebas preliminares de los algoritmos de correlación usando imágenes de color. La metodología de prueba se basó en los siguientes pasos.

1. Se codificaron a nivel software, los métodos descritos en 3.4.2.
2. Se probaron los algoritmos modificados usando imágenes estéreo rectificadas sintéticas.

Las imágenes resultantes muestran de manera cualitativa, mejoras en las estimaciones de profundidad. La figura 4.14 resume los resultados obtenidos por la simulación en hardware y los obtenidos por las implementaciones en software de los métodos descritos en 3.4.2. También se presentan los errores promedios por píxel MPE. La representación tridimensional de los mapas de profundidad ayuda para una mejor visualización.

La figura 4.14 muestra únicamente los resultados de los métodos 1 y 3, con sus respectivos espacios de color, usando el par estéreo Tsukuba.

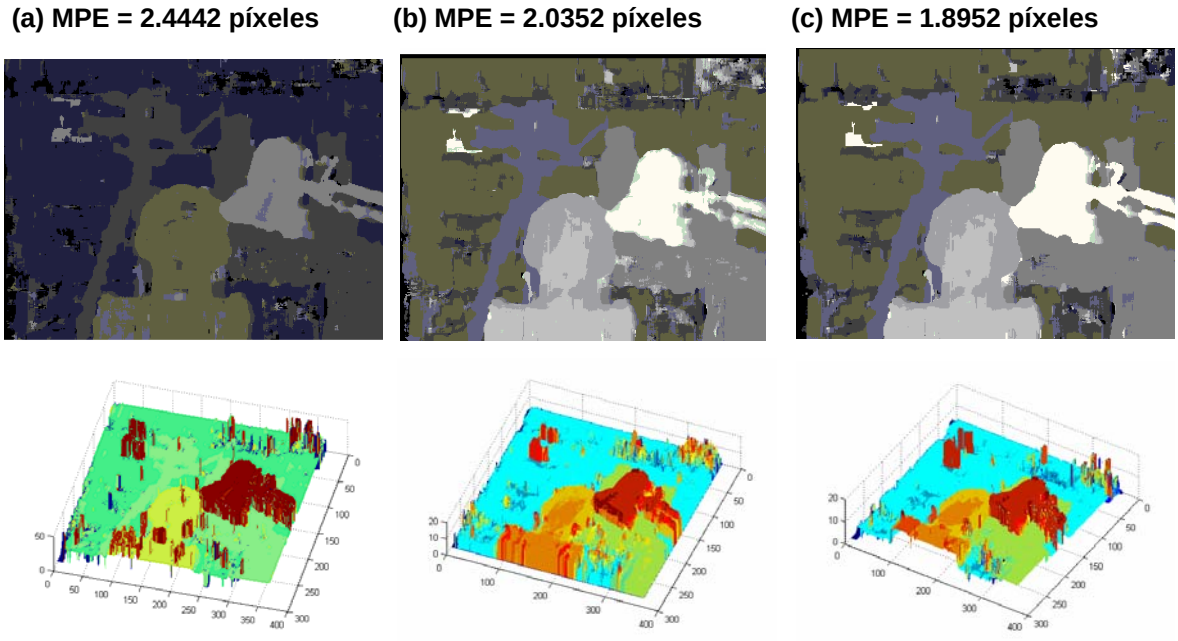

Figura 4.14 Mapas de disparidad y de profundidad. De izquierda a derecha: (a) Simulación HW, (b) método 1 con XYZ, (c) método 3 con $H_1H_2H_3$.

Obsérvese que la métrica MPE muestra que la mejor estimación fue usando la correlación con el método 3 y el espacio de color $H_1H_2H_3$.

De la misma manera que se explicó en la sección anterior, la extensión a la arquitectura estéreo requiere un mayor análisis, no de rendimiento, sino, un análisis de implementación en hardware.

4.4 Resultados de síntesis.

En la sección 4.2, se establece que los resultados de procesamiento, así como los de síntesis, se generaron a nivel simulación. En el caso de los

resultados de síntesis, se muestran los resultados obtenidos por el paquete de desarrollo DK4 de Celoxica. Debido que los resultados de procesamiento (mapas de disparidad y transformaciones geométricas) han sido validados en un nivel de simulación, se establece como resultados de síntesis correctos, los generados en el mismo nivel de simulación.

La herramienta DK4 permite la descripción de hardware, usando el lenguaje Handel-C que tiene sintaxis similar a la del lenguaje C. Una vez generada una descripción de hardware, DK4 provee un analizador sintáctico, el cual pasa su resultado a un compilador de alto nivel. El proyecto que contiene el diseño de hardware se puede configurar para realizar simulación o para generar archivos EDIF, entre otras configuraciones. Si se configura como EDIF, el compilador entrega resultados preliminares de síntesis, es decir, el número de compuertas equivalentes, el número de Flip-Flops utilizados por el diseño, así como una estimación del periodo mínimo del reloj del diseño. Son estos resultados, los reportados en esta sección. Para describir el espacio ocupado por el diseño dentro del dispositivo, se hará en términos de flips-flops (FFs). Cabe recordar que 1 CLB contiene 4 slices, y un slice contiene 2 FFs.

4.4.1 Síntesis del módulo Transformación Geométrica.

En la sección 3.4.2 se describió el funcionamiento del módulo *Transformación Geométrica* y su ruteador. El mapeo de coordenadas de un espacio a otro, realmente sólo requiere la reescritura de los píxeles de una localidad a otra. Debido a esto, el diseño del módulo *Transformación Geométrica* es relativamente sencillo y pequeño.

Los resultados arrojados por el compilador de DK4 con la configuración EDIF, se muestran a continuación. La estimación del área usada se muestra en la tabla 4.1.

info:*Area estimation*: After compilation

***FFs*:**	**68**
Block memory bits:	0
Distributed memory bits:	0
NANDs:	1058
LUTs:	0
Logic tiles:	0
ALUs:	0
Other gates:	0

Tabla 4.1 Estimación de área del módulo Transformación Geométrica.

El número de FFs arrojados es de 68, nótese que este número no representa ni el 1% de la capacidad del dispositivo FPGA (10,944 FFs). Dado que en realidad, el módulo *Transformación* Geométrica sólo realiza un ruteo de datos, el espacio usado por el módulo es aceptable. Es de esperarse, que la extensión a este módulo de procesamiento, al adaptar la interpolación bilineal, cambie significativamente el uso de área del dispositivo.

El resultado del análisis temporal del módulo *Transformación Geométrica* generado por DK4 indica que el tiempo mínimo requerido es de 8.597ns.

El módulo *Transformación geométrica* tarda 3 ciclos de reloj para reescribir un píxel en su posición correcta. Como la frecuencia máxima del módulo es de 116.4 MHz, se puede calcular que para una imagen de tamaño 640x480 píxeles, se requiere de 0.0079 segundos, por lo que este módulo procesa 126 fps (cuadros por segundo).

4.4.2 Síntesis del módulo Correlación estéreo.

La descripción gráfica del módulo de correlación estéreo se presenta en la figura 3.6. El módulo está integrado por procesadores fila mas componentes para calcular el índice del valor mínimo del vector *winSAD*. Los procesadores fila a su vez, se componen de elementos procesador que calculan el valor SAD de dos ventanas. Para calcular el número de procesadores fila necesarios para cumplir con los requerimientos del sistema, es necesario realizar un análisis de rendimiento, con el fin de determinar cuantos índices de disparidad se requieren al finalizar el procesamiento para cumplir con la velocidad de video. Para realizar este cálculo también es necesario conocer la frecuencia máxima del reloj.

Se inicia el análisis partiendo de los resultados de implementación del elemento procesador. La tabla 4.2 muestra los resultados para 1, 2 y 3 elementos procesador trabajando con el ruteador y el control del módulo *Correlación estéreo.*

La herramienta DK4 arroja como resultados de síntesis de alto nivel los siguientes:

info:Area estimation: After compilation	***1 PE***	***2 PE***	***3 PE***
***FFs*:**	**1,171**	**1,380**	**1,588**
Block memory bits:	476	493	510
Distributed memory bits:	51	51	51
NANDs:	127,165	198,430	269,632
LUTs:	0	0	0
Logic tiles:	0	0	0
ALUs:	0	0	0
Other gates:	0	0	0

Tabla 4.2 Estimación de área para 1, 2 y 3 PE.

Se observa de la tabla 4.2 que la diferencia del número de flip-flops usados para 1, 2 y 3 elementos procesador es 208. Es decir, la síntesis de alto nivel de DK4 establece el uso de 208 FFs por cada PE y 964 para el ruteador y el control. Con esta información se puede suponer que un procesador fila se implemente con 964+(208x9) FFs, es decir 2,836 FFs. La tabla 4.3 muestra los resultados de la herramienta DK4 al sintetizar un procesador fila.

info:*Area estimation*: After compilation

***FFs*:**	**2,836**
Block memory bits:	612
Distributed memory bits:	51
NANDs:	696844
LUTs:	0
Logic tiles:	0
ALUs:	0
Other gates:	0

Tabla 4.3 Estimación de área para 1 rowProcessor.

La tabla 4.3 corrobora la suposición de que cada elemento procesador requiere 208 FFs para su implementación en hardware. La frecuencia máxima para el reloj de la arquitectura es de 66.1 Mhz.

De acuerdo a los resultados de síntesis obtenidos, se puede graficar el número *k* de procesadores fila contra el porcentaje de uso de recursos en dispositivo FPGA XC4VFX12. Por otro lado, la figura 3.13 muestra el número de ciclos que requieren *k* procesadores fila para procesar una imagen de 640x480 píxeles. La gráfica 4.15 que representa el porcentaje de uso en el FPGA complementa la gráfica 3.13.

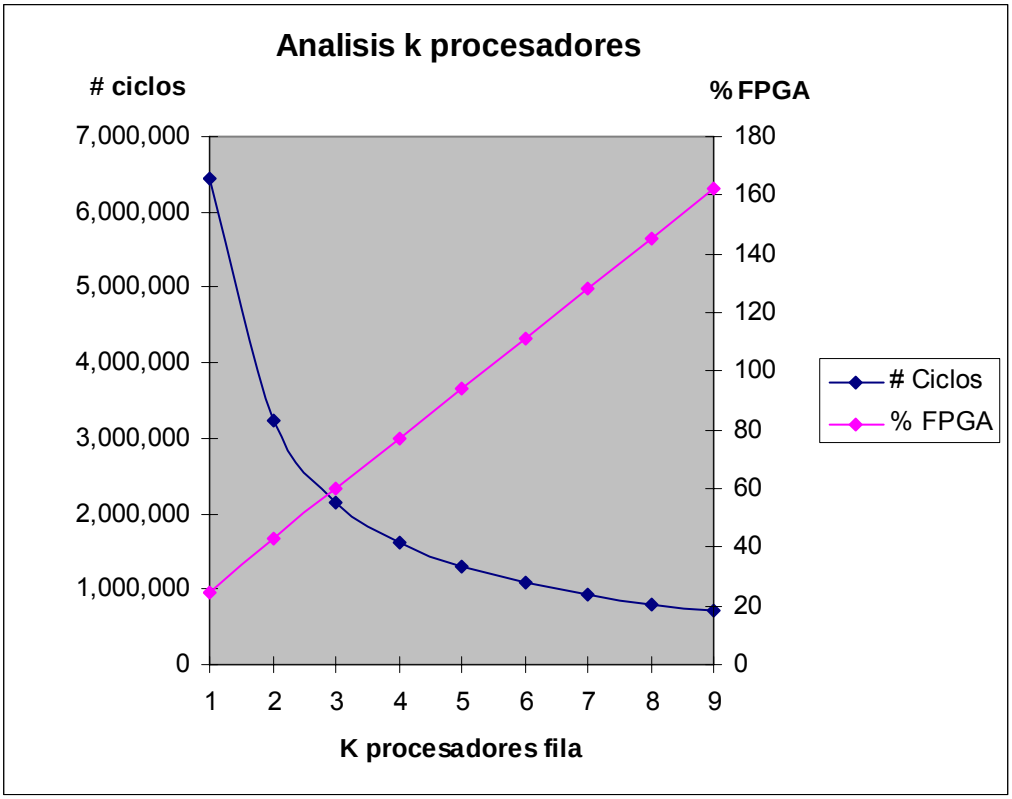

Figura 4.15 Análisis K procesadores fila

Se observa en la figura 4.15 que la curva que representa el número de ciclos necesarios para concluir el procesamiento con *k* procesadores se intersecta con la curva del porcentaje de uso en el FPGA, con *k* = *3*.

Por otro lado, la tabla 4.4 muestra los cálculos que establecen la velocidad de procesamiento para *k* procesadores. La tabla 4.4 complementa el análisis para determinar el valor *k*, es decir, el número de procesadores fila que balancea el compromiso entre velocidad de procesamiento y uso de recursos en el FPGA.

1	2	3	4	5	6	7	8	9	10
Procesadores fila	Resolución	Frecuencia Mhz.	1 frame en *x* seg.	30 frames en *x* seg.	fps	PE	FPGA	% FPGA	# Mpps
1	640x480	66.1	0.10	2.93	10.25	9	XC4VFX12	25.78	3.148
2	640x480	66.1	0.05	1.46	20.49	18	XC4VFX12	42.88	6.294
3	640x480	66.1	0.03	0.98	30.74	27	XC4VFX12	59.99	9.443
4	640x480	66.1	0.02	0.73	40.98	36	XC4VFX12	77.09	12.589
5	640x480	66.1	0.02	0.59	51.23	45	XC4VFX12	94.20	15.737
6	640x480	66.1	0.02	0.49	61.48	54	XC4VFX140	9.64	18.886
7	640x480	66.1	0.01	0.42	71.72	63	XC4VFX140	11.12	22.032
8	640x480	66.1	0.01	0.37	81.97	72	XC4VFX140	12.60	25.181
9	640x480	66.1	0.01	0.33	92.22	81	XC4VFX140	14.08	28.329
12	1 megaP.	66.1	0.03	0.98	30.74	108	XC4VFX140	18.50	32.233
48	4 megaP.	66.1	0.03	0.98	30.74	432	XC4VFX140	71.85	128.932

Tabla 4.4 Análisis de implementación.

Descripción de la tabla 4.4 por columnas:

1. Número de procesadores fila en la arquitectura estéreo.
2. Tamaño de las imágenes estéreo procesadas.
3. Frecuencia máxima en Mhz. de la arquitectura estéreo.
4. Tiempo en segundos para generar un mapa de disparidad.
5. Tiempo en segundos para generar 30 mapas de disparidad.
6. Cuadros por segundo. fps.
7. Número de elementos procesador implementados.
8. Dispositivo FPGA.
9. Porcentaje de uso en el dispositivo de la columna 8.
10. Número de megapíxeles procesados por segundo.

Obsérvese las dos últimas filas de la tabla 4.4, las cuales establecen que implementando 12 y 48 procesadores fila se alcanza velocidad de video usando imágenes de resolución 1 y 4 megapíxeles respectivamente. Este punto demuestra la escalabilidad de la arquitectura estéreo propuesta. Si

bien es cierto que el número de FFs incrementa de manera considerable, también es cierto, que el dispositivo FPGA XC4VFX12 que contiene 10,944 FFs es el dispositivo más chico de su familia (precio estimado: $331.00 usd [3]). El fabricante Xilinx [31] muestra las características de la familia Virtex 4, siendo el dispositivo XC4VFX140 el que más número de FFs tiene disponibles (pero también el más costoso, precio estimado: $2,640.00 usd [3]), esto es 126,336 FFs. Dicho dispositivo es capaz de implementar la arquitectura estéreo de tiempo real para imágenes de 4 megapíxeles.

La tabla 4.4 muestra en la columna 10, el número de píxeles procesados por *k* procesadores fila. Usando estos datos, la figura 4.16 muestra las curvas para *k* procesadores fila, el número de ciclos que requieren y el número de píxeles procesados por segundo.

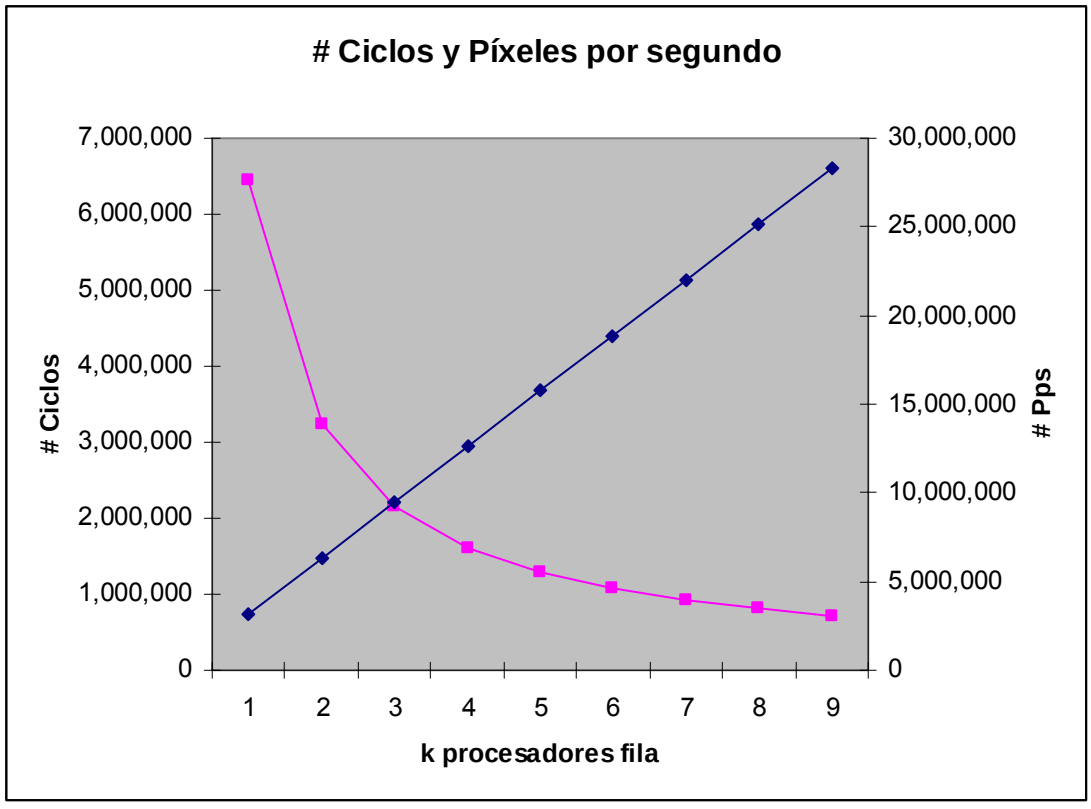

Figura 4.16 Píxeles por segundo.

A partir de las gráficas 4.15 y 4.16, se deduce que implementado 3 procesadores fila se alcanza velocidad de video para imágenes de tamaño 640x480 píxeles ocupando 60% del área del dispositivo FPGA de desarrollo. En base al análisis anterior, se muestran los resultados de la síntesis de alto nivel generados por la herramienta DK4 para la arquitectura estéreo con 3 procesadores fila. La tabla 4.5 muestra la información de ocupación de área implementando 3 procesadores fila en el FPGA XC4VFX12 cuya frecuencia máxima de 66.1 Mhz.

info:*Area estimation*: After compilation

***FFs*:**	**6580**
Block memory bits:	918
Distributed memory bits:	51
NANDs:	2008720
LUTs:	0
Logic tiles:	0
ALUs:	0
Other gates:	0

Tabla 4.5 Estimación de área para la arquitectura estéreo con 3 rowProccessor.

Los resultados de la síntesis de alto nivel, corroboran los datos de la tabla 4.4. Por lo tanto, se concluye que la implementación que balancea de mejor manera los compromisos entre velocidad de procesamiento y área ocupada en el FPGA es con 3 procesadores fila *(k=3)*, la cual alcanza velocidad de video para imágenes de tamaño 640x480 píxeles ocupando 60% del área del dispositivo FPGA de desarrollo.

Se seleccionó la arquitectura hardware descrita en [2] para compararse con la arquitectura de este trabajo en base a los siguientes puntos:

1. Arquitectura hardware basada en FPGA
2. Cálculo de disparidad a través de métodos de correlación por área.

3. Arquitectura con los puntos 1 y 2 más reciente.

La tabla 4.6 muestra la comparación entre el sistema estéreo presentado en [2] contra la arquitectura descrita en este documento. Las figuras 4.17 y 4.18 muestran gráficamente la comparación de la tabla 4.6.

	VRSV [2]	Arquitectura estéreo en esta investigación
# FPGAs	4	1
# FFs totales	153,600	10,944
# FFs usados	83,026	6,580
Frecuencia máxima MHZ	50	66.1
Tamaño de la imagen	256x360	480x640
Cuadros por segundo	30	30.74
# Mpps	2.764	9.443

Tabla 4.6 Comparación entre el sistema VRVS y la arquitectura en esta investigación.

La tabla 4.7 muestra las arquitecturas hardware encontradas durante la revisión del avance científico junto con los datos de la arquitectura propuesta en este documento. La columna con la métrica *pps* sirve para comparar todas las arquitecturas. Nótese que el número de píxeles procesados por segundo reportado por la arquitectura de este documento es el número alcanzo con la configuración de 3 procesadores fila, sin embargo, aumentando el número de procesadores fila se aumenta los *pps* también.

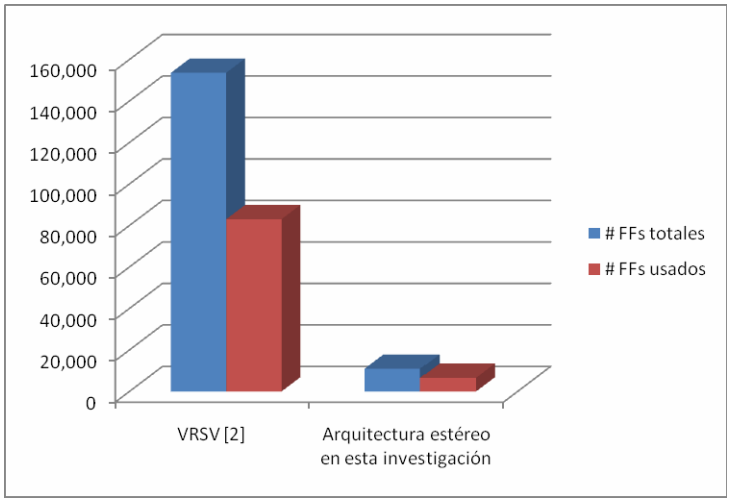

Figura 4.17 Comparación de flips-flops disponibles y usados por VRSV y la arquitectura en esta investigación.

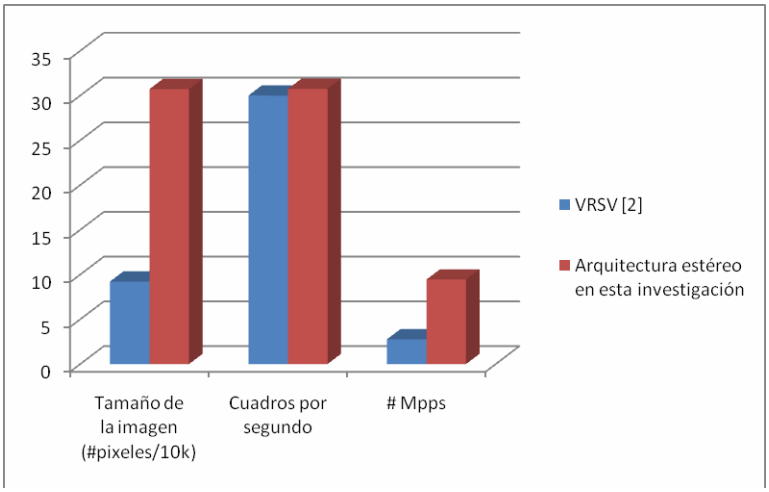

Figura 4.18 Comparación de Mpps, fps y tamaño de las imágenes procesadas por VRSV y la arquitectura en esta investigación.

Arquitectura HW	Tamaño de la imagen	Velocidad de procesamiento	Medida de simili-tud	Procesador	Cámaras	#píxeles por se-gundo
INRIA 1993	256x256	3.6 fps	NCC	Perle-1	3	235,930
CMU iWarp 1993	256x240	15 fps	SSAD	64Processor iWarp2	3	921,600
JPL 1995	256x240	1.7 fps	SSD	Datacube	2	104,448
CMU 1995	256x240	30 fps	SSAD	Custom HW y DSP	6	1,843,200
Point Grey Triclops 1997	320x240	6 fps	SAD	PII 450 MHz.	3	460,800
SRI SVS 1997	320x240	12 fps	SAD	TSM3920 DSP	2	921,600
PARTS engi-ne 1997	320x240	42 fps	Census	FPGA	2	3,225,600
CSIRO 1997	256x256	30 fps	Census	FPGA	2	1,966,080
SAZAN 1999	320x240	20 fps	SSAD	FPGA	9	1,536,000
Point Grey Triclops 2001	320x240	13 fps	SAD	PIV 1.4GHz.	3	998,400
SRI SVS 2001	320x240	30 fps	SAD	PII 700MHz.	2	2,304,000
Xicotencatl. 2000 [30]	240x240	85 fps	SAD	FPGA	2	4,896,000
VRSV 03 [2]	256x360	30 fps	LWPC	4 FPGAs	2	2,764,800
Bumblebee XB3 2006 [19]	1280x960	15 fps	SAD	PIV 2.0GHz. 512RAM 64AGP	2	18,432,000
DeepSea G2 Stereo Vision 2007 [27]	512x480	30 fps	Census	FPGA, DSP, PowerPC, DeepSea	2	7,372,800
García V. 2008	**640x480**	**30 fps**	**SAD**	**FPGA**	**2**	**9,216,000**

Tabla 4.7 Arquitecturas Hardware de visión estéreo actualizada.

4.5 Discusión.

Durante el capítulo 4, se han mostrado los resultados de toda la investigación realizada durante el trabajo de tesis. Se ha propuesto, diseñado e implementado una arquitectura basada en FPGA para la recuperación tridimensional en tiempo real, que al agregarle los controladores de periféricos necesarios, se obtendrá una cámara inteligente 3D.

Además, se han propuesto y probado, adaptaciones a la arquitectura estéreo que permiten una mejor estimación de los mapas de disparidad. Análisis preliminares a los métodos de extensión demuestran la factibilidad de mejorar los resultados generados por la arquitectura estéreo, sin embargo, se debe profundizar en los análisis para determinar si la adaptación a la implementación en hardware es posible o no.

Los resultados de implementación descritos en la sección 4.4 demuestran la escalabilidad de la arquitectura estéreo con la finalidad de establecer parámetros de búsqueda de acuerdo a la aplicación final. El crecimiento de la arquitectura estéreo no compromete la frecuencia máxima del sistema, pues al trabajar en paralelo los procesadores fila, solo se requiere replicarlos para obtener tantos índices de disparidad como sean requeridos para alcanzar la restricción de velocidad de video (en realidad, la arquitectura estéreo puede ser configurada para alcanzar velocidades iguales o mayores a 30 fps).

La arquitectura hardware propuesta en este trabajo de investigación puede ser comparada usando como métrica el número de píxeles procesados por segundo. La tabla 2.4 muestra las arquitecturas encontradas en la literatura, obsérvese que la implementación comercial *BumbleBee XB3* [19] es la que procesa mayor número de píxeles por segundo, esto es, alrededor de 18 megapíxeles por segundo. La arquitectura propuesta en este trabajo

puede alcanzar y superar los 18 megapíxeles por segundo al agregar 6 o más procesadores fila.

Es necesario resaltar que los resultados obtenidos son a nivel simulación, por lo que es necesario generar los controladores para periféricos como sensores, memorias, dispositivos de comunicación, entre otros, con el fin de lograr resultados a un nivel mayor a la simulación. Resolver estos problemas técnicos se proponen como trabajo a futuro.

Capítulo 5

Conclusiones.

5.1 Conclusiones

Durante este trabajo, se realizó una investigación sobre los algoritmos de correspondencia estéreo, así como los algoritmos para las correcciones y transformaciones geométricas. El análisis permite seleccionar los algoritmos más adecuados para una aplicación específica. También la investigación permite establecer el estado del avance científico con el fin de proponer una arquitectura novedosa, la cual permite ser configurada con diferentes valores de tamaño de imágenes estéreo, tamaño de ventana de búsqueda, tamaño de rango de disparidad manteniendo la velocidad de video. Otro punto que nace de la investigación, es la propuesta de extensión a la arquitectura estéreo con el fin de mejorar los resultados arrojados por la misma.

Los siguientes puntos resumen los objetivos planteados y alcanzados en esta tesis:

- Analizar el algoritmo que resuelve la corrección geométrica de un par de imágenes estéreo y adaptarlo a una arquitectura hardware.

Se diseñó e implementó el módulo de transformación geométrica que resuelve la rectificación de un par de imágenes estéreo.

- Analizar un algoritmo que resuelva la correlación por área de un par de imágenes estéreo y adaptarlo a una arquitectura hardware.

- Analizar diversos algoritmos para el cálculo de medidas de similitud y adaptar los seleccionados a una arquitectura hardware reconfigurable.

Se diseñó e implemento el módulo para la correlación estéreo aplicando la medida de similitud SAD.

- Analizar, diseñar e implementar un modelo hardware que resuelva la gestión de accesos a memoria de manera eficiente.

Se diseñaron e implementaron módulos ruteadores que permiten el acceso a la memoria explotando el ancho de banda de las mismas.

- Analizar, diseñar e implementar un modelo hardware que sincronice los modelos anteriores

Se diseñó e implementó el modulo para el control del flujo de datos para la arquitectura estéreo.

En resumen, las conclusiones son:

- La corrección radial y cualquier transformación geométrica aplicada a imágenes digitales se puede implementar en hardware mediante tablas de mapeo.
- La rectificación estéreo tiene mejores resultados aplicando interpolación a nivel subpíxel.
- Estadísticamente, la mejor medida de similitud usando correlación estéreo basada en área es la suma de diferencias absolutas (*SAD*).
- El cálculo de mapas de disparidad a través de la correlación estéreo tiene mejores resultados usando imágenes codificadas en color que usando imágenes codificadas en niveles de gris.

- El mejor compromiso entre velocidad de procesamiento y uso de cursos del FPGA con el que se trabajó (XC4VFX12) es con una configuración: tamaño de la imagen de 480x640, rango de búsqueda de disparidad de 15 píxeles y con una ventana de búsqueda de 7x7 píxeles es implementando 3 procesadores fila, cada uno con 9 elementos procesadores.
- La arquitectura estéreo es multiconfigurable para alcanzar cualquier velocidad de procesamiento teniendo como límite el tamaño de recursos en el FPGA usado.

5.2 Aportaciones

La principal contribución de esta investigación es el diseño y la implementación a nivel simulación de una arquitectura hardware capaz de ser configurada con diferentes parámetros, como rango de búsqueda de disparidad, tamaño de ventana de búsqueda, tamaño de las imágenes estéreo. Cualquier configuración alcanza la restricción de tiempo real al implementar *k* procesadores fila. Los siguientes puntos resumen las contribuciones en este trabajo de investigación:

- Diseño e implementación de transformaciones geométricas mediante tablas de mapeo, las cuáles aceleran la velocidad de procesamiento y reducen significativamente el uso de recursos en el FPGA.
- Diseño e implementación del conjunto de elementos procesadores llamado *Procesador fila* para el cálculo de disparidades en un par de imágenes estéreo. El *Procesador fila* hace uso de datos independientes con respecto al tiempo para acelerar el algoritmo logrando el pro-

cesamiento paralelo que permite aumentar la velocidad de procesamiento y reducir el uso de recursos en el FPGA.

- Diseño de una arquitectura hardware que permite resolver el problema de correspondencias usando imágenes codificadas en algún espacio de color, logrando aumentar la calidad de los mapas de disparidad.

5.3 Trabajo futuro.

En la sección 3.4.1 se realiza una breve investigación sobre tópicos que permitan mejorar la calidad de las imágenes transformadas por modelos matemáticos geométricos. Se concluye que la interpolación bilineal es la sugerida para realizar una extensión al módulo *Transformación Geométrica*. Una segunda técnica de interpolación se sugiere en la sección 3.4.2, en esta ocasión, la interpolación sugerida es lineal y permitirá aumentar la resolución del mapa de disparidad, creando mapas de disparidad con resolución a nivel subpíxel.

Durante la investigación del avance científico, se reviso el trabajo [2], el cual demuestra estadísticamente que los mapas de disparidad generados por los algoritmos de correspondencia, en particular, los basados en correlación por área, pueden mejorar las estimaciones al hacer uso de imágenes estéreo codificadas en algún espacio a color. Debido a esto, se propone realizar una extensión a la arquitectura estéreo propuesta con el fin de implementar los métodos descritos en [2]. En la sección 3.4.2, se realiza un análisis preliminar para determinar la factibilidad de esta extensión.

La figura 3.11 muestra el diagrama a bloques de la arquitectura global, más aún, muestra componentes hardware que serían necesarios implementar con el fin de intercomunicar la arquitectura con dispositivos periféricos.

Los periféricos propuestos son los controladores para los sensores, así como para algún dispositivo USB que permita a la arquitectura estéreo interactuar con algún dispositivo externo, como lo puede ser una computadora personal. Por lo tanto, se propone diseñar e implementar los componentes hardware que permitan controlar los sensores de imágenes así como al dispositivo USB.

La arquitectura hardware propuesta en este trabajo de investigación puede ser integrada dentro de un sistema más complejo, para realizar diversas tareas, como la navegación autónoma de vehículos, medición de partes industriales, entre otros.

Referencias.

[1] Abdel-Aziz Y. I. and Karara H. M. **"Direct linear transformation into object space coordinates in close-range photogrametry."** In *Proc. Symp. Close-Range Photogrametry*, pp. 1--18, University of Illinois, Urbana, 1971.

[2] Ahmad D. **"Video-Rate Stereo Vision on Reconfigurable Hardware".** Tesis de Maestría. Universidad de Toronto. 2003

[3] Avnet Electronics Marketing. **Price List**. URL: www.ptgrey.com. Agosto 2007.

[4] Banks J, and Corke P., **"Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision,"** In *Int'l J. Robotics Research*, vol. 20, no. 7, 2001.

[5] Belli T., Cord M. and Philipp S. **"Colour contribution for stereo image matching"**. In *First International Conference on Colour in Graphics and Image Processing CGIP'2000",* Saint-Etienne, France 2000.

[6] Bramberger M., Brunner J., Rinner B., Schwabach H. **"Real-time video analysis on an embedded smart camera for traffic surveillance".** In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*. 2004

[7] Bramberger M., Doblander A., Maier A. Rinner B. **"Distributed Embedded Smart Cameras for Surveillance Applications".** In *IEEE Computer Magazine*. pp. 68-76. Febrero 2006.

[8] Chambon S. and Crouzil A. **"Color stereo matching using correlation measures."** In *Complex Systems Intelligence and Modern Technological Applications - CSIMTA 2004*, p. 520-525, Cherbourg, France, September 2004.

[9] Faugeras 0., Hotz B., Matthieu B., Vieville T., Zhang Z., Fua, Theron, Moll, Berry, Vuillemin, Bertin, and Proy C., **"Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications,"** INRIA Technical Report 2013, 1993.

[10] García V. **"Arquitectura Hardware basada en FPGA para la corrección radial de imágenes digitales".** Tesis de Licenciatura. BUAP. Puebla, México. 2006.

[11] González R. and Word R. **"Digital image processing"**. USA: Addison-Wesley, 2002.

[12] Haralick R. and Shapiro L. **"Computer and Robot Vision. Vol. 2,"** USA: Prentice Hall, 2002.

[13] Jian X., Malcolm A. and Zhongping F. **"Camera Calibration with Micron Level Accuracy."** SIMTech Technical Report (AT/01/037/PS), 2001.

[14] Kersten T. and Baltasavias E. **"Sequential estimation of sensor orientation for stereo images sequences."** In *International Archives of Photogrammetry and Remote Sensing* 1994, Vol. 30, Part 5, pp. 206-213.

[15] Leeser M., Miller S., Yu H. **"Smart camera based on reconfigurable hardware enables diverse real-time applications".** In *Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04)*. 2004.

[16] Myron Z., Darius B. and Gregory D. **"Advances in Computational Stereo."** In *IEEE Transactions on pattern analysis and Machine Intelligence*, 2003. vol.25 no.8. p. 993-1008.

[17] Nourbakhsh I., Andre D., Tomasi C. and Genesereth M. **"Obstacle Avoidance Via Depth From Focus".** In *ARPA Image Understanding*. Workshop 1996. Stanford, CA.

[18] Poelzeitner W., Ulm M. **"A comparative study of camera calibration algorithms with application to spacecraft navigation".** In *SPIE* 1994. Vol. 2350, Videometrics III, pp. 187-196

[19] Point Gray Research. **"BumbleBee Stereo Vision Camera Systems"**. Datashet. URL: www.ptgrey.com. Noviembre 2006.

[20] Rosas M. **"Rectificación radiométrica y geométrica de un sistema estereo".** Tesis de Maestría, INAOE. Tonanzintla, México. 2006.

[21] Scharstein D., Szeliski R. **"A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms".** Technical Report MSR-TR-2001-81. 2001.

[22] Smith S.M. and Brady J.M. **"SUSAN - a new approach to low level image processing"**. In *Int. Journal of Computer Vision*, 23(1):45--78, May 1997.

[23] Truco E., Verri A. **"Introductory techniques for 3-D computer vision".** USA: Prentice Hall, 1998.

[24] Tsai R. **"A versatil camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses".** In *IEEE Journal of robotics and automation* 3(4): 323-334. 1987

[25] Venkateswar V. and Chellappa R., **"Hierarchical Stereo and Motion Correspondence Using Feature Groupings".** In *Int'l J. Computer Vision*, vol. 15, pp. 245-269, 1995.

[26] Wolf W., Ozer B., Lv T. **"Smart cameras as embedded systems**" In *IEEE Computer Magazine*. pp. 48-53, Septiembre 2002.

[27] Wodfill J., Gordon G., Jurasek D. Brown T., Buck R. **"The Tyzx DeepSea G2 Vision System, A Taskable, Embedded Stereo Camera".** In *Proceedings of the IEEE Computer Society Workshop on Embedded Computer Vision*, Conference on Computer Vision and Pattern Recognition, (New York, NY), June 2006

[28] Woodfill J. and Von Herzen B., **"Real-Time Stereo Vision on the PARTS Reconfigurable Computer".** In *Proc. IEEE Workshop FPGAs for Custom Computing Machines*, pp. 242-250, 1997.

[29] Woodham R., Iwahori Y., Barman R. **"Photometric Stereo: Lambertian Reflectance and Light Sources with Unknown Direction and Strength".** Technical Report 91 - 18. Canadian Institute for Advanced Research. 1991.

[30] Xicotencatl J. **"Arquitectura Hardware de visión estéreo en tiempo real".** Tesis de Maestría. INAOE. Tonanzintla, México. 2000.

[31] Xilinx Inc. **Xilinx Virtex4 User Manual**. URL: www.xilinx.com. Abril 2007.

[32] Zabih, R. And J. Woodfill. **“Non-parametric Local Transforms for Computing Visual Correspondence”**. In *Proceedings of 3rd European Conference on Computer Vision*. May 1994. pp. 150-158.