UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA

Fusión de secuencias de vídeo de alta velocidad procedentes de cámaras desplazadas espacial o temporalmente

Alumno: David Otero García

Tutor: Jesús Bescós Cano

SEPTIEMBRE 2010

PALABRAS CLAVE

Mapas de profundidad, mapas de disparidad, puntos homólogos, homografías, *matching*, par estéreo, estereovisión, *frame rate*, *pinhole*, líneas epipolares, *scanlines*, *baselines*.

RESUMEN

El principal objetivo de este PFC es la implementación de un algoritmo que sea capaz de generar una única secuencia de video a partir de imágenes grabadas por diferentes cámaras. La peculiaridad de las imágenes grabadas es que son imágenes de una escena con profundidad. De ahí que el proceso de generar la secuencia de video no sea trivial.

Como resultado se ha obtenido un método nuevo que obtiene dicha secuencia de vídeo de manera bastante aceptable, teniendo en cuenta los inconvenientes insalvables que tiene adjunto este proceso.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor, Jesús Bescós, por permitir que realizase este Proyecto Fin de Carrera, por su apoyo, dedicación y sobre todo por su paciencia durante todos estos años de estudios. También a la gente de VPULab en especial a Marcos y Fabri por su inestimable ayuda.

Gracias a mi familia por sus consejos y todo el amor y el apoyo que me han dado durante toda mi vida, por los valores que me han inculcado y por hacer de mí una persona principalmente feliz.

Gracias a mis compañeros de universidad por haber sido un pilar fundamental a la hora de realizar mis estudios. Porque sin ellos hubiera sido muy difícil. Gracias a Eduardo, Gonzalo, Marta, Pepe, Fernando, Juanma, Cris… por todos los momentos de risas, de estudio, de prácticas, de viajes, de abrazos y de mucho cariño, siempre os recordaré.

Gracias también a todos los profesores con los que he coincidido en estos años, su conocimiento ha servido para instruirme y han fomentado el interés por aprender que sigo teniendo cada día de mi vida.

Gracias a Silvi, Henry y al Cabo por estar siempre conmigo en los malos momentos.

Muchas gracias.

David Otero,

Septiembre 2010.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

Este proyecto se basa en el uso de cámaras de alta velocidad que permiten obtener miles de imágenes por segundo. Las aplicaciones de estos sistemas son variadas: balística, estudios de automoción, visualización de explosiones, biomecánica, anuncios de publicidad, formula 1, reacciones químicas…

Debido al elevado coste de este tipo de cámaras, tanto mayor cuanto mayor es su tasa de cuadro o *frame rate*, este proyecto explora la posibilidad de utilizar varias cámaras, situadas en posición lo más parecida posible pero capturando cuadros o imágenes en instantes distintos, para luego fusionar las secuencias resultantes de cada cámara e imitar así la captura de una sola cámara que operara a mayor velocidad o resolución temporal.

El caso más directo sería el uso de cámaras con sus líneas de visión paralelas y lo más próximas entre sí.

Si se utilizaran dos cámaras, colocadas una paralela a la otra tanto como permita su carcasa, y capturando imágenes alternativamente obteniendo así dos secuencias, una por cada cámara, la fusión consistente en reproducir alternativamente una imagen de cada cámara resultaría en que la escena mostrada no tendría nada que ver con la real: tendríamos diferentes iluminaciones, puntos ocultos, descuadre de la imagen…

El objetivo es desarrollar las técnicas necesarias para corregir las imágenes capturadas de modo que todas parezcan provenir de una misma cámara. Dejando aparte aspectos de iluminación (suponemos cámaras idénticas y lo suficientemente próximas como para que la iluminación las afecte por igual), Para ello vamos a utilizar una técnica bien conocida llamada homografía. La cual permite, mediante una operación matemática y partiendo de ciertos parámetros, convertir la imagen obtenida por una cámara dada en la imagen correspondiente que se grabaría desde otra cámara diferente.

Para escenas que carecen de profundidad (un cuadro en una pared por ejemplo) el cálculo de la homografía que relaciona la imagen de cada cámara con una de referencia es una técnica robusta que resuelve el problema. El problema viene cuando la escena en cuestión tiene profundidad (por ejemplo un aula vista desde atrás, con sus alumnos, pupitres, pizarra etc.). Por una parte, las imágenes procedentes de cada cámara ya no están relacionadas por una homografía. De hecho,

existen puntos ocluidos, puntos que desde una cámara se ven pero desde las otras no.

El objetivo de éste proyecto es desarrollar una técnica que permita obtener resultados aceptables en una escena con profundidad capturada por dos cámaras próximas La aproximación inicial será identificar en ambas imágenes capas de puntos situados a igual profundidad para luego aplicar a cada capa la homografía que le corresponda. Para ello trataremos los principales problemas involucrados: detección de profundidades, cálculo de homografías, y resolución de oclusiones.

1.2. OBJETIVOS

El principal objetivo de éste proyecto es obtener una única secuencia de imágenes partiendo de dos grabaciones de una escena determinada. La escena grabada será característica por tener profundidad. La secuencia obtenida debe parecer una grabación hecha por una única cámara.

La disposición de las cámaras será de forma que se simule una visión estéreo. Así la cámara izquierda no podrá grabar los puntos que estén demasiado a la derecha y viceversa para la cámara derecha, pero en cambio los punto del medio (los de la derecha para la cámara izquierda y los de la izquierda para la cámara derecha) serán comunes. El resultado será una secuencia aproximada de la situación grabada.

Inicialmente se deben aplicar técnicas de emparejamiento de puntos homólogos (matching) para poder determinar la homografía que transforma las imágenes entre las dos cámaras. Esto sería la detección de puntos característicos en la primera imagen (primera cámara) y ser capaz de detectar los mismos puntos en la segunda (segunda cámara).

Posteriormente la idea sería tratar de obtener mapas de profundidad de dos imágenes, para poder diferenciar las profundidades de la imagen real. La relación entre las imágenes de un mismo plano en dos cámaras distintas viene dada por una homografía; así nuestro objetivo es separar la escena en varias profundidades y posteriormente tratar cada profundidad como un plano independiente y aplicarle la homografía adecuada. Por ello el hallar los mapas de profundidad de la escena es vital.

Una vez obtengamos los mapas de profundidad de las imágenes debemos ver de que forma extraer dichas profundidades o capas. Posteriormente se estudiara el proceso de aplicar las homografías por capas viendo sus resultados, problemas y posibles soluciones.

1.3. ESTRUCTURA DE LA MEMORIA

El documento que documento que describe este Proyecto Fin de Carrera se organiza de la siguiente manera:

Capítulo 2: En este capítulo se presenta la planificación general del proyecto y el estado del arte de los procesos analizados. Así como la teoría necesaria para comprender el funcionamiento de los distintos métodos realizados.

Capítulo 3: Este capítulo esta dedicado íntegramente a la fase de detección y emparejamiento de puntos característicos. Se describe la implementación de los métodos utilizados y se analizan los resultados obtenidos.

Capítulo 4: En este capítulo se trata el tema de la disparidad. Como obtener las distintas profundidades de una escena. Para ello se implementa un el método escogido y se analizan los resultados.

Capítulo 5: En este capítulo se desarrolla el algoritmo principal. Se hacen uso de los resultados anteriores y se integran para concluir con el objetivo del proyecto. Se presentan distintos casos y se analizan los resultados.

2. PLANTEAMIENTO Y ANTECEDENTES[1]

2.1. PLANTEAMIENTO DEL OBJETIVO

Para poder entender el desarrollo del proyecto y presentar los posibles problemas asociados lo mejor es plantear el objetivo final como una serie de pasos y poder extraer distintas conclusiones en cada uno de estos pasos. Para ello veamos la Figura 1 que nos ayudará a visualizar el proceso.

[1] En éste capítulo se hace referencia principalmente a [5], [6], [15], [25] y [17]. Se han extraído textos parciales y algunas imágenes.

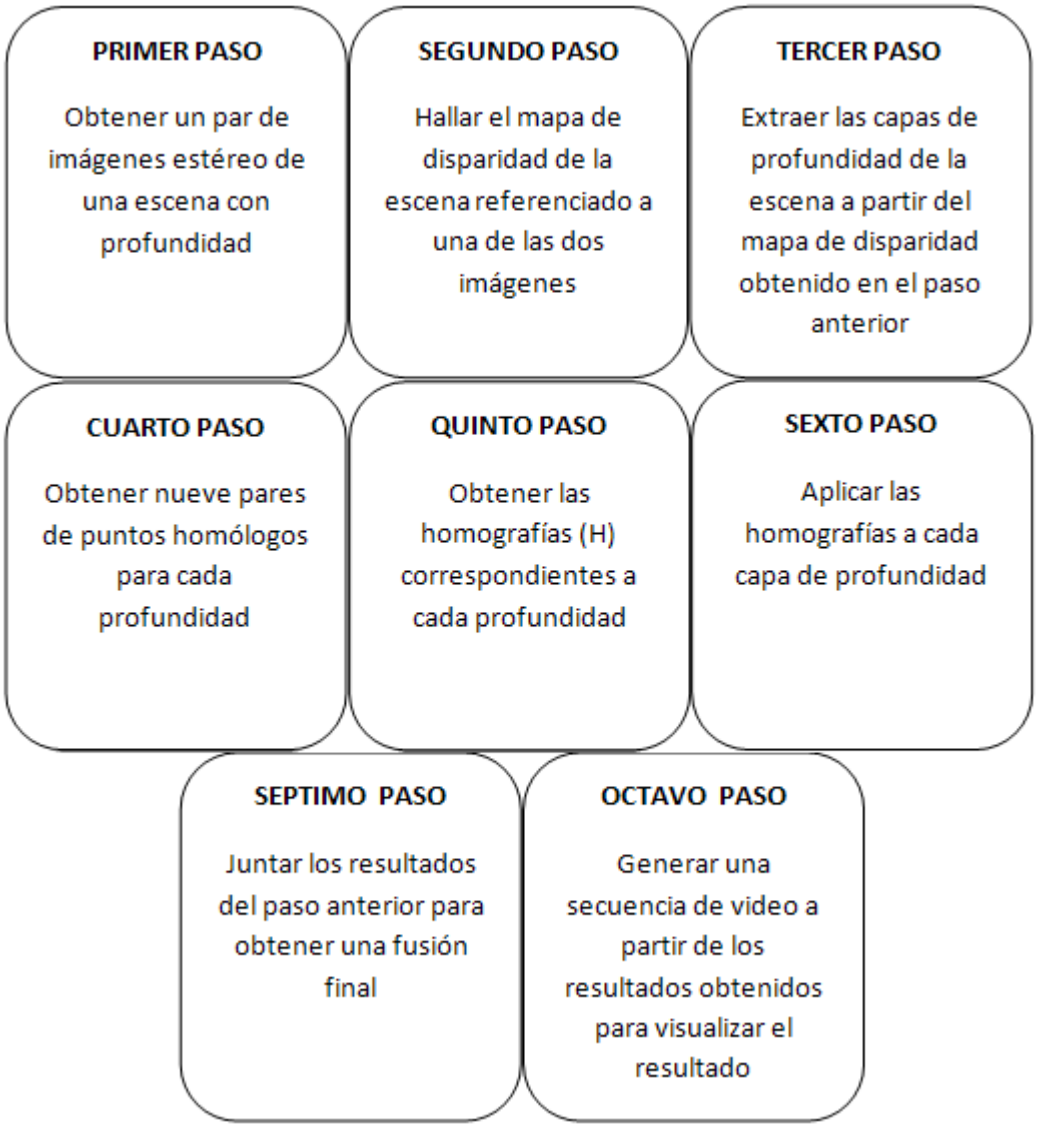

FIGURA 1 : ESQUEMA GENERAL DEL FUNCIONAMIENTO DE NUESTRO ALGORITMO

El primer paso no corresponde estrictamente al ámbito de éste proyecto pero es necesario obtener de alguna manera un par de imágenes con las que posteriormente se trabajará. Por ello los problemas relacionados con la obtención de las imágenes no se detallaran, tales como el calibrado de las cámaras, rectificado de las imágenes etc. Son problemas que se tendrán en cuenta para explicar algunos de los resultados pero que no se van a tratar de resolver.

El proyecto se puede dividir en cuatro grandes bloques. El primero sería hallar el mapa de disparidad referente a una de las imágenes dadas. Este corresponde con el paso dos. El segundo bloque trataría de obtener los pares de puntos homólogos entre dos imágenes dadas. El cuarto paso sería la ubicación de este bloque. Y el tercer bloque se encargaría de encontrar las homografías (H) de cada capa, sería el paso quinto. Por último el cuarto bloque se encargaría de gestionar los resultados de

los bloques anteriores y utilizarlos para conseguir la fusión final, es decir, el paso tercero, sexto, sétimo, y octavo.

Para tener una visión todavía más amplia se adjunta la Figura 2 donde se muestra el funcionamiento general del proceso por bloques.

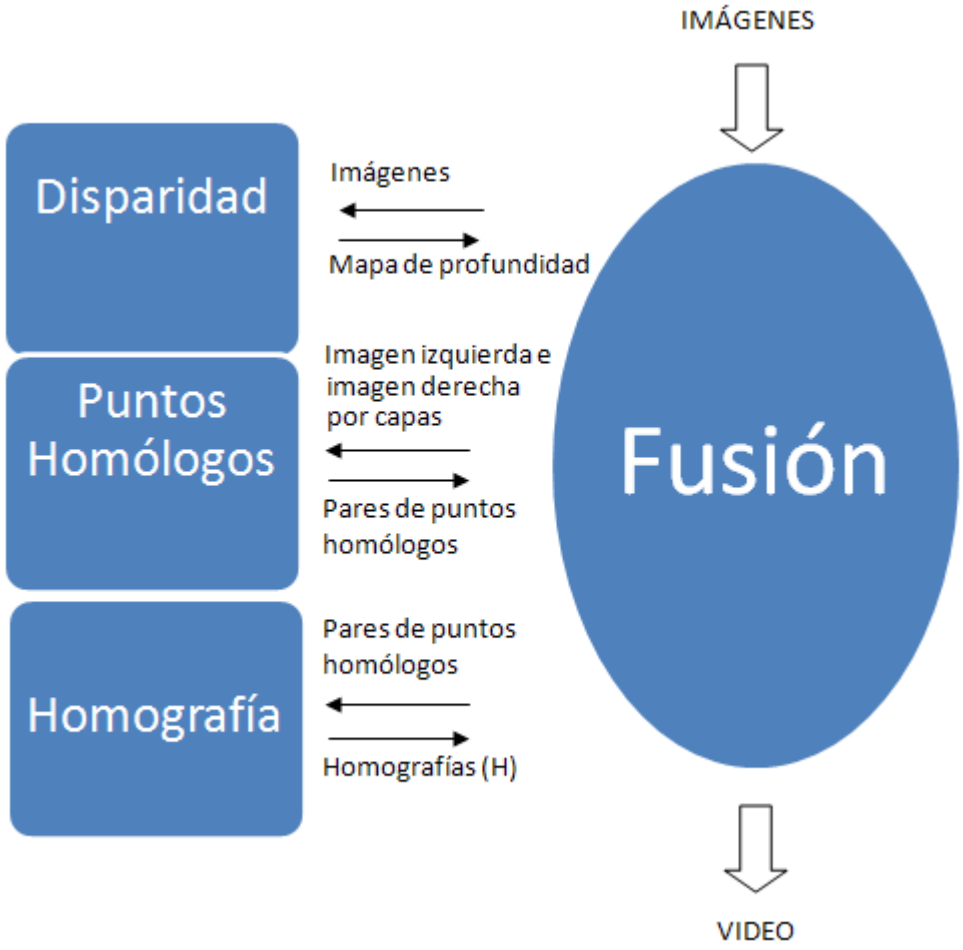

FIGURA 2 : DIAGRAMA GENERAL DEL FUNCIONAMIENTO DE NUESTRO ALGORITMO

Se puede observar en la Figura 2 que el desarrollo de los bloques de *disparidad*, de los *puntos homólogos* y de las *homografías* son independientes. Es decir, se pueden desarrollar de manera separada y evaluar los resultados de manera separada, una vez sean aceptables podemos incluirlos en el algoritmo final. Para el resultado final no son independientes pues se necesita tanto de los mapas de disparidad, como de los emparejamientos de puntos homólogos y las homografías por capas para poder realizar la fusión final y obtener el resultado global. Se puede observar en las relaciones entre bloques los requisitos que necesita cada bloque y los resultados que devuelve cada uno.

El bloque de *disparidad* necesita el solamente el par de imágenes estéreo para generar el mapa de profundidad.

El bloque de *puntos homólogos* necesita la imagen izquierda y cada capa de profundidad de la imagen derecha (que serán extraídas en el bloque *fusión*) para poder establecer los emparejamientos de puntos homólogos.

El bloque de *homografía* necesitará los pares de puntos homólogos extraídos en el bloque anterior para poder calcular la homografía (H) de cada capa.

El bloque de disparidad, el de *puntos homólogos y el de fusión* son más extensos en comparación con el bloque de *homografías*. Por ello el bloque de *homografías* será integrado en el bloque de *fusión* y se verá su aplicación en el capítulo 5. Para el desarrollo del proyecto nos viene igual empezar por uno que por otro. La decisión ha sido empezar por la detección automática de puntos homólogos que se verá en el capitulo tercero. Posteriormente se atenderá el problema de hallar el mapa de profundidad de la escena. Y por último se desarrollará el algoritmo general para hallar la fusión total de los resultados obtenidos.

En este segundo capítulo se tratara de explicar la teoría necesaria para entender las características de cada proceso, así como los diferentes métodos que se pueden adoptar y los posibles problemas asociados. Para desarrollar el proyecto hemos dicho que empezaremos por el bloque sobre los puntos homólogos. Ahora bien, para comprender de forma teórica del proceso entiendo que es necesario comenzar por el tema principal, las homografías. Por esto en este capitulo empezaremos por ver el problema general sobre las homografías. Para ello se necesita introducir una parte teórica para poder entender los mecanismos que se adoptan en el proceso de realizar una homografía. Posteriormente veremos la teoría y posibles soluciones referentes al bloque sobre los puntos homólogos. A continuación trataremos el tema de la disparidad y finalmente se concluirá exponiendo de antemano los posibles problemas asociados según los métodos escogidos.

2.2. HOMOGRAFÍAS Y SU CÁLCULO

2.2.1. TRANSFORMACIONES PROYECTIVAS

En esta lección introducimos las principales ideas geométricas y notaciones que serán necesarias para la comprensión de las siguientes secciones. En particular se introducen la geometría de las transformaciones proyectivas del plano. Estas transformaciones modelan la distorsión geométrica que se introduce sobre un plano cuando se toma una imagen del mismo con una cámara de perspectiva.

Bajo una cámara de perspectiva, algunas propiedades geométricas se conservan, tales como la colinealidad (una línea recta se proyecta en una recta), mientras otras propiedades no, por ejemplo el paralelismo: en general las líneas paralelas no se presentan como tales en la imagen. La geometría proyectiva modela el proceso de adquisición de la imagen al mismo tiempo que da una representación matemática apropiada para los cálculos.

2.2.2. EL PLANO PROYECTIVO 2D

Como es bien conocido un punto en el plano puede ser representado por un par de coordenadas (x,y) en R^2. Por lo que es común identificar el plano con R^2. Considerando R^2 como un espacio vectorial, el par de coordenadas (x,y) es un vector, es decir un punto se identifica como un vector. A continuación introduciremos la notación homogénea para puntos y líneas del plano.

2.2.2.1. VECTORES FILA Y COLUMNA

Más adelante nos interesará considerar aplicaciones lineales entre espacios vectoriales y representar dichas aplicaciones como matrices. De forma usual el producto de una matriz y un vector es otro vector. Esto nos conduce a distinguir entre vectores fila y vectores columna ya que una matriz puede ser multiplicada por un vector columna por la derecha y por un vector fila por la izquierda. Las entidades geométricas serán representadas por defecto como columnas. Los símbolos tales como **x**, siempre se representan con un vector columna. De acuerdo a esto un punto de un plano estará representado por un vector columna $\mathbf{x}=(x,y)^T$.

2.2.2.2. PUNTOS Y LÍNEAS

Una línea en el plano se representa por una ecuación tal como *ax+by+c=0*. Por tanto, una línea puede representarse de forma natural por un vector $(a,b,c)^T$. La correspondencia entre líneas y vectores $(a,b,c)^T$ no es uno-a-uno, ya que las líneas *ax+by+c=0* y *(ka)x+(kb)y+(kc)=0* son la misma para cualquier constante k distinta de cero. De hecho, dos vectores relacionados por una escala global son considerados como equivalentes. Una clase de equivalencia de vectores bajo esta relación se conoce como un vector homogéneo. El conjunto de clases de equivalencia de vectores en R^3 –$(0,0,0)^T$ forma el espacio proyectivo P^2. Esta notación –$(0,0,0)^T$ indica que el vector $(0,0,0)^T$, no corresponde a ninguna línea, está excluido.

2.2.2.3. REPRESENTACIÓN HOMOGÉNEA DE PUNTOS

Un punto $\boldsymbol{x}=(x,y)^T$ está sobre la línea $\boldsymbol{l}=(a,b,c)^T$ si y solo si *ax+by+c=0*. Esta ecuación puede escribirse como un producto interno de vectores, $(x,y,1)(a,b,c)^T =$

$(x,y,1)\boldsymbol{l}$ o de la siguiente manera $(x,y,1)\boldsymbol{l} = 0$, es decir, el punto $\boldsymbol{x}=(x,y)^T$ ha sido representado como un 3-vector añadiendo una tercera coordenada 1. Debemos notar que para cualquier constante k distinta de cero, se seguirá verificando la misma igualdad $(kx,ky,k)\boldsymbol{l} = 0$. Es por tanto natural considerar el conjunto de vectores $\boldsymbol{(kx,ky,k)}^T$ para distintos valores de k como una representación del punto $(x,y)^T$ en R^2. Por tanto al igual que con las líneas, los puntos se representan por vectores homogéneos. Un vector arbitrario homogéneo representante de un punto es de la forma $\boldsymbol{x}=(x_1,x_2,x_3)^T$, representando el punto $(x_1/x_3, x_2/x_3)^T$ en R^2. Tanto puntos, como vectores homogéneos, son también elementos de P^2.

Del resultado anterior se deduce que un punto $\boldsymbol{x}$ esta sobre una línea $\boldsymbol{l}$ si y solo si se verifica $\boldsymbol{x}^T\boldsymbol{l}=0$.

Es evidente que para especificar un punto habrá que dar dos valores: sus coordenadas x e y. De igual manera para especificar una línea habrá que dar dos parámetros (los dos cocientes independientes (a:b:c)) y por tanto tan solo tiene dos grados de libertad.

2.2.2.4. INTERSECCIÓN DE LÍNEAS

Dadas dos líneas $\boldsymbol{l}=(a,b,c)^T$ y $\boldsymbol{l}'=(a',b',c')^T$ queremos calcular su intersección. Definimos el vector $\boldsymbol{x}= \boldsymbol{l}\times\boldsymbol{l}'$, donde × representa el producto vectorial. De la identidad del producto escalar triple $\boldsymbol{l}'\cdot(\boldsymbol{l}\times\boldsymbol{l}')=\boldsymbol{l}\cdot(\boldsymbol{l}\times\boldsymbol{l}')=0$ puede verse que $\boldsymbol{l}^T\boldsymbol{x}=\boldsymbol{l}'^T\boldsymbol{x}=0$. Por tanto si consideramos a $\boldsymbol{x}$ un representante de un punto, dicho punto estará sobre ambas rectas $\boldsymbol{l}$ y $\boldsymbol{l}'$, y por tanto en la intersección de ambas.

Este resultado muestra que el punto intersección de dos retas $\boldsymbol{l}$ y $\boldsymbol{l}'$ está dado por $\boldsymbol{x}=\boldsymbol{l}\times\boldsymbol{l}'$.

De igual manera que el resultado anterior, se puede deducir que el vector que define la recta que pasa por dos puntos $\boldsymbol{x}$ y $\boldsymbol{x}'$ está dado por $\boldsymbol{l}=\boldsymbol{x}\times\boldsymbol{x}'$.

2.2.2.5. PUNTOS IDEALES Y RECTA DEL INFINITO

Uno de los aspectos más engorrosos del estudio de la geometría en el caso euclídeo es la necesidad de distinguir constantemente entre puntos del infinito y puntos finitos. Por ejemplo, en el caso de la intersección de rectas paralelas se conoce que no tiene solución en el caso euclídeo y se dice que se intersecan en el infinito. La geometría proyectiva, gracias a la notación en coordenadas homogéneas, permite abordar el estudio de propiedades de intersección de puntos y rectas sin necesidad de hacer distinción entre los casos finito o infinito.

Ya hemos visto antes que es posible expresar la intersección de dos rectas como el producto cruzado de dos vectores, por tanto podemos asegurar que siempre existirá una solución independientemente de la situación relativa de las rectas. Es de interés analizar cual será la solución para el caso de rectas paralelas. Puede verse sin dificultad que en estos casos el vector homogéneo que representa la solución tendrá obligatoriamente su tercera coordenada igual a cero, lo que corresponde a un punto fuera del plano R^2. Por tanto podemos decir que los puntos finitos de R^2 están representados por vectores de R^3 con tercera coordenada $x_3 \neq 0$. Los puntos de R^3 con tercera coordenada igual a cero, $x_3=0$, se denominan puntos ideales o puntos del infinito. Notaremos que el conjunto de los puntos ideales, $(x_1,x_2,0)^T$, esta todo sobre una recta que llamaremos recta del infinito y cuyo vector es $\boldsymbol{l}_\infty=(0,0,1)^T$.

2.2.3. TRANSFORMACIONES DEL PLANO PROYECTIVO

Una proyectividad del plano es una aplicación invertible de P^2 en P^2 (es decir de 3-vectores homogéneos) que aplica líneas a líneas. De forma más precisa:

Definición. Una proyectividad en una aplicación invertible h de P^2 en P^2 tal que tres puntos x_1, x_2, y x_3 están alineados si y solo si $h(x_1)$, $h(x_2)$ y $h(x_3)$ lo están.

Las proyectividades forman un grupo ya que es un conjunto cerrado para la operación inversa y la composición. Una proyectividad también se denomina colineación u homografía.

Un importante resultado que permite usar las propiedades algebraicas de una proyectividad es el siguiente:

Teorema. Una aplicación h de P^2 en P^2 es una proyectividad si y solo si existe una (3x3)-matriz no singular (una matriz es singular si y solo si su determinante es cero) H tal que para cualquier punto en P^2 representado por un vector $\boldsymbol{x}$ es verdad que $h(x)=Hx$.

El teorema asegura que cualquier proyectividad puede representarse como una transformación lineal invertible en coordenadas homogéneas y que, inversamente, cualquier transformación de este tipo es una proyectividad.

Como consecuencia de este teorema se puede dar la siguiente definición alternativa:

Definición. (Transformación proyectiva). Una transformación proyectiva entre planos es una transformación lineal sobre 3-vectores homogéneos $\boldsymbol{x}$, representada por una 3x3-matriz H, $x'= Hx$.

Es importante resaltar que dado el carácter homogéneo de los vectores, la matriz *H* puede multiplicarse por una constante sin que la transformación se modifique. Por tanto la matriz *H* también es de tipo homogéneo y esta definida salvo una constante de proporcionalidad.

Como consecuencia la matriz *H* tan solo posee 8 elementos independientes, ya que uno de ellos lo fija la constante de proporcionalidad.

Una transformación proyectiva transforma un plano en otro plano equivalente en el que se conservan todas las propiedades invariantes a las proyectividades

Ejemplo de Proyectividad. Consideremos una proyección entre planos que se produce al hacer pasar todos los rayos que salen de los puntos de uno plano por un punto común (el centro de proyección). Observar la Figura 3.

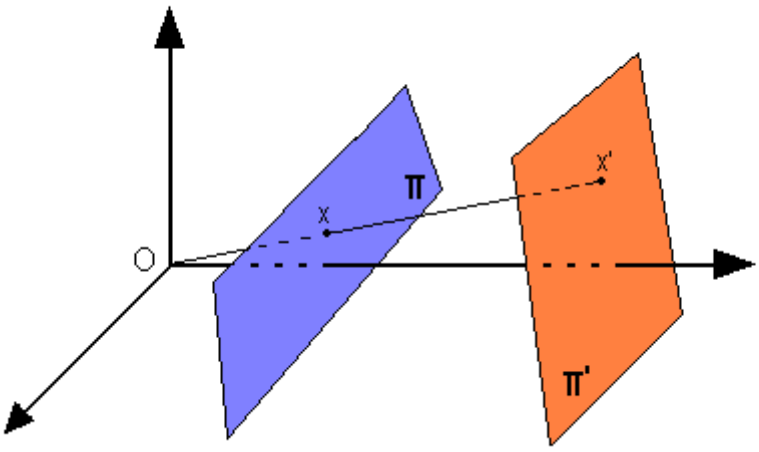

FIGURA 3 : EJEMPLO DE PROYECTIVIDAD

Es evidente que este sistema de proyección conserva las líneas, en el sentido que una línea sobre un plano se transforma en una línea del otro plano. Después de los resultados anteriores es evidente que si se definen sistemas de coordenadas en cada plano y los puntos se expresan en coordenadas homogéneas, el sistema de proyección central se puede expresar por $x' = Hx$ siendo H una 3x3-matriz invertible.

Si los sistemas coordenados de ambos planos son Euclídeos entonces la transformación se llama de perspectiva y puede verse que solo depende de seis grados de libertad.

Efecto de la perspectiva. La formas se distorsionan bajo el efecto de la perspectiva, así por ejemplo las formas que conocemos son cuadriláteros rectangulares se presentan como cuadriláteros deformados. Ver la Figura 4: Efecto de la perspectiva .

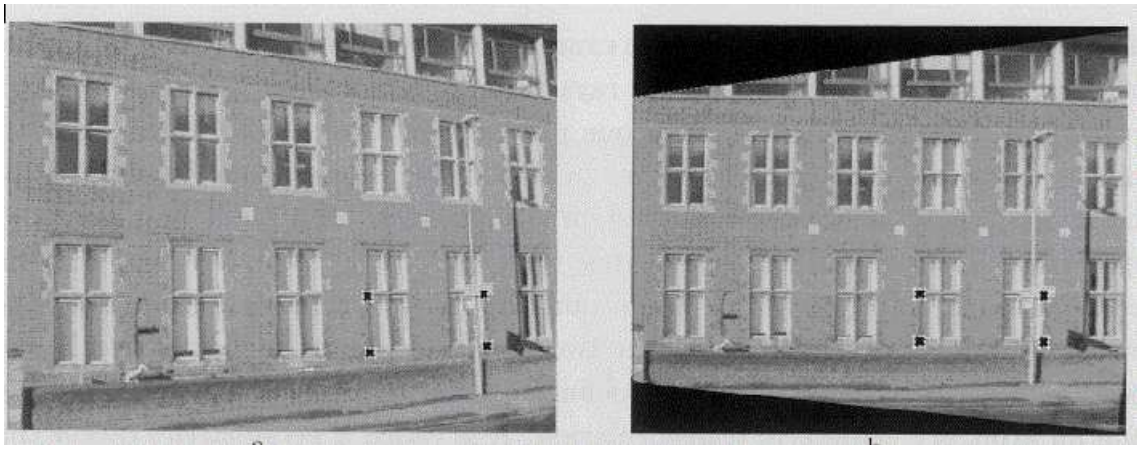

FIGURA 4: EFECTO DE LA PERSPECTIVA

En general las líneas paralelas no se conservan paralelas. En el ejemplo anterior hemos visto que la proyección central se puede considerar un caso particular de las transformaciones proyectivas y por tanto se puede expresar como el resultado de una transformación lineal invertible. Es posible por tanto deshacer la deformación calculando la transformación inversa y aplicándosela a la imagen. Las imágenes siguientes muestran el proceso en cuestión, donde se muestra la nueva imagen sintetizada con las deformaciones corregidas.

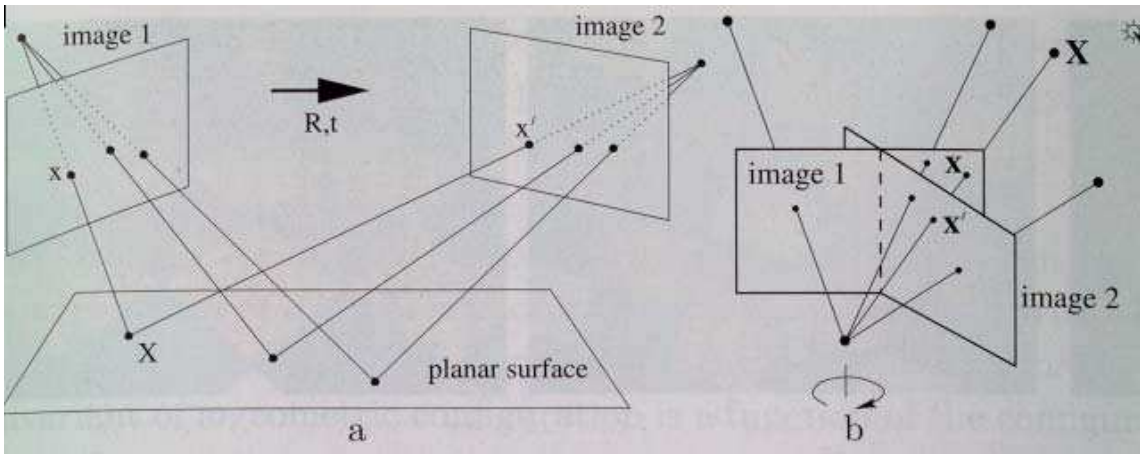

FIGURA 5: PROCESO DE DEFORMACIÓN DE LAS IMÁGENES

FIGURA 6: EJEMPLO DEL PROCESO DE DEFORMACIÓN

El estudio de cómo estimar la transformación inversa a partir de puntos de ambas imágenes se llevará a cabo en las siguientes secciones.

2.2.4. JERARQUÍA DE TRANSFORMACIONES

Dentro del grupo de las transformaciones proyectivas, o grupo lineal proyectivo, existen numerosos subgrupos de gran interés que vamos a ir estudiando.

El grupo de las matrices *nxn* invertibles con elementos reales es el *grupo lineal general* sobre *n* dimensiones o *GL(n).* Para obtener el *grupo lineal proyectivo* es necesario identificar la clase de matrices que son equivalentes salvo por una constante; este grupo se nota *PL(n)* (es un grupo cociente de *GL(n)*). En nuestro caso *n*=3.

Los subgrupos importantes de *PL(3)* incluyen el *grupo afín*, que es el subgrupo de *PL(3)* consistente en las matrices para las cuales la última fila es (0,0,1), el *grupo euclídeo,* que es un subgrupo del grupo afín para el cual la submatriz 2x2 superior-izquierda es ortogonal. También se puede identificar el *grupo euclídeo orientado* en el caso en que el determinante de la submatriz es igual a 1.

A continuación iremos estudiando estas transformaciones comenzando por las más especializadas.

2.2.4.1. CLASE I: ISOMETRÍAS

Las isometrías son transformaciones del plano R^2 que preservan distancia euclídea. Una isometría se representa por

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \left[\begin{array}{c|c|c} \varepsilon\cos\theta & -sen\theta & t_x \\ \varepsilon sen\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{array}\right] \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \text{Ec. 1}$$

donde ε=±1. Si ε=1 entonces la isometría preservará la orientación y es una transformación euclídea. Si ε=-1 la isometría invertirá la orientación (una reflexión).

Las transformaciones euclídeas modelan los movimientos de los cuerpos rígidos. Son las isometrías más importantes y serán objeto de nuestro estudio, pero las isometrías que invierten la orientación también aparecen como posibles ambigüedades en el proceso de recuperación de la estructura geométrica a partir de imágenes.

Una transformación euclídea plana se puede escribir de forma concisa como sigue

$$x' = H_E x = \left[\begin{array}{c|c} R & t \\ 0^T & 1 \end{array}\right] x \qquad \text{Ec. 2}$$

Donde R es un matriz de rotación 2x2 (es decir ortogonal tal que $R^T R = R R^T = I$), t es un 2-vector de traslación y 0 es un 2-vector nulo. Casos de especial interés son una rotación pura (cuando t=0) y una traslación pura (cuando R=I). A las transformaciones euclídeas también se les conoce como *desplazamientos*.

Una transformación euclídea entre planos tiene tres grados de libertad, uno para la rotación y dos para la traslación. Por tanto habrá que estimar tres parámetros para definir la transformación. Ya que en la ecuación de la transformación cada correspondencia entre puntos fija dos ecuaciones lineales entre los elementos de la matriz, en este caso con solo dos correspondencias entre puntos sería posible calcular los parámetros de la transformación.

Los invariantes de este grupo son bien conocidos: longitudes, ángulos y áreas.

2.2.4.2. CLASE II: SEMEJANZAS

Una *semejanza* es una isometría compuesta con un escalado isotrópico. En el caso de una representación euclídea compuesta con una escala, la semejanza tiene la siguiente representación

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \left[\begin{array}{cc|c} s\cos\theta & -s\,sen\theta & t_x \\ s\,sen\theta & s\cos\theta & t_y \\ \hline 0 & 0 & 1 \end{array}\right] \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \text{Ec. 3}$$

que puede escribirse de forma más compacta como

$$x' = H_S x = \left[\begin{array}{c|c} sR & t \\ 0^T & 1 \end{array}\right] x \qquad \text{Ec. 4}$$

donde *s* representa el factor de escala. Una semejanza se conoce también como una transformación *equiforme* ya que conserva la forma.

Una transformación de semejanza en el plano tiene 4 grados de libertad (1-escala, 1-giro, 2-traslación), por tanto al igual que la isometría puede ser calculada a partir de la correspondencia entre dos puntos.

Los invariantes de esta transformación se pueden construir a partir de los euclídeos teniendo en cuenta la escala. Así pues, los ángulos son los únicos invariantes euclídeos que permanecen. También puede observarse que los cocientes entre longitudes y entre áreas son invariantes de esta transformación.

2.2.4.3. CLASE III: AFINIDADES

Una transformación afín o una *afinidad* se define como una transformación lineal no singular seguida de una traslación. La representación matricial de la misma es

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \left[\begin{array}{cc|c} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ \hline 0 & 0 & 1 \end{array}\right] \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \text{Ec. 5}$$

o en forma compacta

$$x' = H_A x = \left[\begin{array}{c|c} A & t \\ \hline 0^T & 1 \end{array}\right] x \qquad \text{Ec. 6}$$

siendo A una matriz 2x2 no singular. Una transformación afín en el plano tiene 6 grados de libertad correspondiendo a los seis elementos de la matriz. Por tanto necesitará de tres correspondencias entre puntos para poder ser calculada.

Los dos nuevos grados de libertad aparecen como consecuencia de que en las transformaciones afines se pueden producir deformaciones siguiendo una dirección arbitraria (1 ángulo, 1 parámetro de escala que mide el cociente entre las deformación en la nueva dirección y su ortogonal).

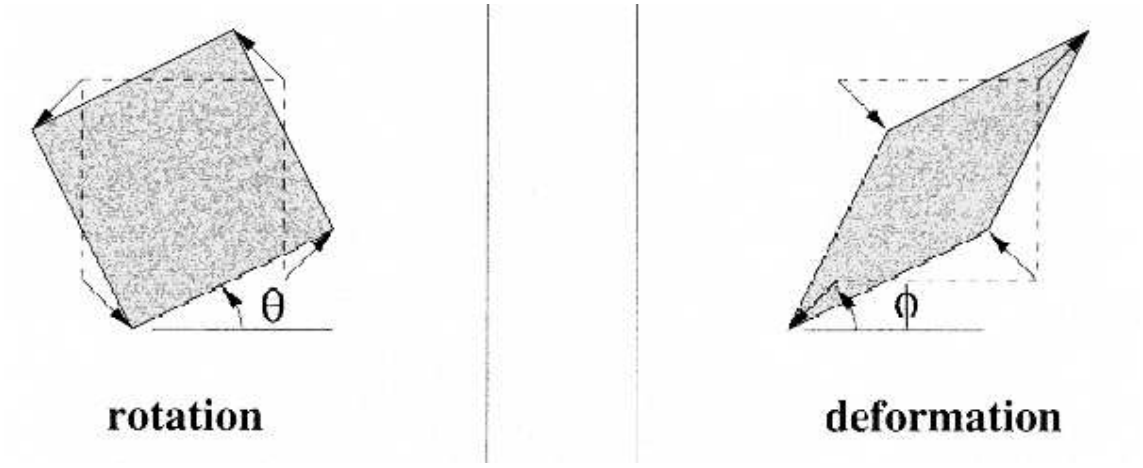

FIGURA 7: DEFORMACIONES AFINES

Dado que una transformación afín permite deformaciones no isotrópicas, los invariantes de las semejanzas no lo serán de las afinidades. Ahora los invariantes más importantes son: El paralelismo, los cocientes de longitudes de segmentos de líneas paralelas, cocientes de áreas.

Una afinidad preservará o no la orientación del plano en función del signo del determinante de la matriz A. Es sabido que $det(A)= \lambda_1\lambda_2$ (λ_1 y λ_2 corresponden a los autovalores de la matriz *A*. Como los autovalores están asociados a las deformaciones producidas en el plano el valor del determinante es el producto de las deformaciones ejercidas), así el signo del determinante dependerá del signo de las deformaciones.

2.2.4.4. CLASE IV: PROYECTIVIDADES

Las transformaciones proyectivas ya han sido definidas como transformaciones lineales no singulares de coordenadas homogéneas. Evidentemente generalizan las transformaciones afines que son la composición de una transformación lineal general no singular de coordenadas no-homogéneas y de una traslación. La notación compacta de una transformación proyectiva es

$$\mathbf{x}'=\mathbf{H}_{\mathrm{P}}\mathbf{x}=\begin{bmatrix}\mathbf{A} & \mathbf{t}\\ \mathbf{v}^{\mathbf{T}} & v'\end{bmatrix}\mathbf{x} \qquad \text{Ec. 7}$$

donde $\boldsymbol{v}=(v_1,v_2)^T$. La matriz tiene nueve elementos pero solo ocho son independientes. Aunque en muchas ocasiones se fija la escala tomando el valor de *v'=1*, esto no es siempre correcto ya que en algunos casos el verdadero valor de *v'* puede ser 0 y por tanto no ser correcto este escalado. Una proyectividad entre planos puede ser calculada a partir de la correspondencia entre cuatro puntos. Ahora, al contrario de las afinidades, no es posible distinguir entre proyectividades que preservan o invierten la orientación.

El invariante más importante de las proyectividades entre planos es la razón cruzada de cuatro puntos alineados, que se define como

$$RC(x_1,x_2,x_3,x_4)=\frac{|\overline{x_1}\,\overline{x_2}||\overline{x_3}\,\overline{x_4}|}{|\overline{x_1}\,\overline{x_3}||\overline{x_2}\,\overline{x_4}|} \qquad \text{Ec. 8}$$

La principal diferencia entre proyectividades y afinidades es el vector $\boldsymbol{v}=(v_1,v_2)^T$ que define la tercera fila de la matriz de la transformación. En las afinidades este vector es fijo e igual a (0,0) y en cambio en las proyectividades puede ser cualquiera del espacio. Este vector es responsable de los efectos no-lineales de la proyectividad. Para ello comparemos la aplicación de un punto ideal $(x_1,x_2,0)^T$ bajo una afinidad y una proyectividad. Podemos ver sin ninguna dificultad que un punto ideal (un punto del infinito) se proyecta bajo una afinidad en otro punto ideal, mientras que bajo una proyectividad se proyecta en un punto cualesquiera del espacio definido por el vector $\boldsymbol{v}=(v_1,v_2)^T$.

2.2.5. DESCOMPOSICIÓN DE UNA PROYECTIVIDAD

Una transformación proyectiva puede ser descompuesta en una cadena de transformaciones donde cada matriz de la cadena representa una transformación más alta en la jerarquía que la anterior.

$$H = H_S H_A H_P = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} K & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ \mathbf{v}^T & v \end{bmatrix} = \begin{bmatrix} A & t \\ \mathbf{v}^T & v \end{bmatrix} \quad \text{Ec. 9}$$

donde A es una matriz no singular dada por $A=sRK+tv^T$ y K es una matriz triangular superior normalizada con $det(K)=1$. Esta descomposición es válida con tal de que $v \neq 0$ y es única si s se toma positivo. Cada una de las matrices H es la esencia de una transformación del tipo indicado por el subíndice.

Por ejemplo en el proceso de rectificación se debe calcular una transformación proyectiva. Así pues, dicha transformación se puede calcular paso a paso a través de la descomposición anterior.

La transformación H_P (2 grados de libertad) mueve el vector $\mathbf{v}=(v_1,v_2)^T$ a la recta del infinito, la transformación H_A (2 g.l.) afecta a las propiedades afines pero no mueve la línea del infinito, la transformación H_S (4 g.l.) es una semejanza general que no afecta ni a las propiedades afines ni a las proyectivas.

2.2.6. ESTIMACIÓN DE PROYECTIVIDADES

Este apartado se centra en el problema de estimación de las proyectividades. Por estimación entendemos el cálculo de los parámetros de las transformaciones a partir de medidas de alguna naturaleza. Con objeto de precisar de forma más concreta los problemas de los que estamos hablando, fijaremos las siguientes situaciones:

1. **Homografías 2D.** Dado un conjunto de puntos x_i en P^2 y un correspondiente conjunto de puntos x'_i en P^2, calcular la transformación proyectiva que aplica los x_i en los x'_i. En la práctica los puntos x_i y x'_i son puntos en dos imágenes (o la misma imagen) en donde cada imagen se considera como un plano proyectivo P^2.

2. **Cámaras de proyección 3D a 2D.** Dados un conjunto de puntos X_i en el espacio 3D y un conjunto de correspondientes puntos x_i en una imagen, encontrar la transformación proyectiva 3D-2D que aplica los X_i en los x_i. Cada proyección 3D-2D es la aplicación llevada a cabo por una cámara proyectiva.

3. Cálculo de la matriz fundamental. Dado un conjunto de puntos x_i en una imagen y los correspondientes puntos x'_i en la otra imagen, calcular la matriz fundamental F consistente con estas correspondencias. La matriz fundamental F, es una matriz singular 3x3 que satisface la ecuación $x'^{T}_i Fx_i = 0$ para todo i.

4. Cálculo del Tensor trifocal. Dado un conjunto de puntos en correspondencia en tres imágenes, $x_i \leftrightarrow x'_i \leftrightarrow x''_i$ calcular el tensor trifocal (Como la matriz fundamental pero para 3 vistas).

Estos problemas tienen muchas características en común y las consideraciones relativas a uno de los problemas son también de interés para los otros. Nos centraremos en el estudio del primero de ellos.

Problema: Dados un conjunto de puntos en correspondencia $x_i \leftrightarrow x'_i$ entre dos imágenes, encontrar la matriz no singular 3x3 tal que $x'_i = Hx_i$ para todo i.

Varios comentarios son de interés antes de establecer métodos de cálculo.

Número de puntos requeridos: La primera cuestión a considerar es cuántos puntos o correspondencias son necesarias para estimar la matriz H. Es posible calcular una cota inferior teniendo en cuenta los grados de libertad de la matriz H y el número de restricciones. Por un lado, es obvio de la propia ecuación de la correspondencia que cada pareja de puntos en correspondencia nos da tres ecuaciones en los parámetros de la matriz H, pero de ellas solo dos son linealmente independientes ya que un punto sólo tiene dos grados de libertad, sus coordenadas. Por tanto, cada correspondencia de puntos fija dos restricciones sobre la matriz H. Por otro lado la matriz H tiene 9 parámetros pero está definida salvo un factor de escala, por lo que sólo tiene 8 parámetros independientes. En conclusión se puede establecer que cuatro correspondencias de puntos son suficientes para calcular la matriz H.

En todo el razonamiento anterior se supone que de los cuatro puntos escogidos no hay grupos de tres puntos que estén alineados.

Soluciones Aproximadas: Acabamos de ver que si se fijan cuatro correspondencias de puntos, es posible obtener una estimación exacta de H. Sin embargo, esto sólo sería válido para el caso de ideal de que no exista ruido sobre las medidas. En la práctica todas las medidas son ruidosas ya que al menos están afectadas por el ruido asociado de la falta de precisión infinita. Por tanto si se tienen más de cuatro puntos no siempre será posible establecer una matriz H válida para todos los puntos. Así pues, habrá que abordar el problema como la búsqueda de la mejor transformación H dados los datos. Esto será llevado a cabo a través de la minimización de una función de costo sobre los parámetros de la matriz H.

Existen dos familias principales de funciones de costo: aquellas que minimizan un error algebraico y aquellas que minimizan una distancia geométrica o estadística sobre la imagen.

Hemos optado por trabajar con el algoritmo mas extendido dentro de la comunidad de visión por computador. Utilizamos el algoritmo DLT para el cálculo de *H*. En el Apéndice 1 se describe el funcionamiento de dicho método para el cálculo de *H*.

2.3. SELECCIÓN DE PUNTOS HOMÓLOGOS

Debemos ser conscientes que en este apartado tenemos dos partes bien diferenciadas. Primero se deben encontrar puntos característicos en la imagen de referencia y posteriormente se deben de buscar en la imagen posterior para establecer los pares de puntos.

Ahora bien, ¿que entendemos por puntos característicos? Podríamos definirlos como elementos con buenas propiedades de textura que resulten por ello fácilmente identificables. Las características que deben reunir estos puntos de interés en cuatro propiedades son: *distinción, unicidad, invariancia* y *estabilidad*.

La **distinción** significa que un punto debe ser diferente de sus vecinos inmediatos. Esto excluye la selección de puntos pertenecientes a áreas uniformes de la imagen o bordes rectilíneos, ya que los distintas partes de un borde rectilíneo no pueden diferenciarse entre sí.

La **unicidad** significa que un punto debería ser distinguible globalmente, es decir, idealmente no debería parecerse a ningún otro punto de la imagen. Para asegurar que los puntos cumplan esta propiedad debería evitarse la selección de elementos que, aunque resulten localmente distinguibles, aparezcan repetidamente en la imagen. Este tipo de puntos repetidos afecta negativamente a los procesos de búsqueda de correspondencias, al provocar situaciones de confusión.

La **invariancia** de un punto se refiere a que la apariencia de éste no debería variar a consecuencia de las distorsiones geométricas o radiométricas que se prevé que puedan ocurrir, debido a las características del objeto y su movimiento o en relación con la iluminación de la escena o el proceso de formación de la imagen.

La **estabilidad** se refiere, por último, a que la apariencia del punto debería ser invariante respecto al punto de vista. Puntos interesantes de la imagen deben corresponder a puntos de interés del objeto. Deben excluirse puntos que resultan del cruce de bordes de distintos objetos o de un objeto y el fondo, por ejemplo.

Existen procedimientos automáticos para seleccionar puntos con buenas propiedades de distinción y unicidad. Sin embargo asegurar el cumplimiento de las dos últimas propiedades es mucho más dificultoso y requeriría un buen conocimiento a priori del objeto, la escena y el tipo de movimiento esperado. En general, los procedimientos de selección automática de puntos característicos buscan puntos con buena distinción, basándose en alguna medida de variabilidad local de la imagen o en alguna característica como la calidad de esquina o borde.

2.3.1. SELECCIÓN Y DETECCIÓN DE PUNTOS CARACTERÍSTICOS

El problema al que nos enfrentamos sería el de detectar un numero ‘x’ de emparejamientos de puntos homólogos.

El primer paso, como se ha comentado anteriormente, es buscar los puntos de la imagen que aporten información realmente importante. En general, estos puntos denominados de interés o característicos, tienen en común una serie de propiedades. Algunas de ellas son, por ejemplo, que se pueden encontrar de una manera sencilla y formalizable. Es decir, al someter la imagen a una operación algorítmica determinada, dichos puntos destacan significativamente. Además, tienen una posición muy bien definida y el conjunto de píxeles vecinos que hay alrededor del punto característico aporta una gran cantidad de información local relevante.

La propiedad más importante es su estabilidad frente a perturbaciones locales y globales. Éstas pueden incluir transformaciones simples como rotaciones y traslaciones, o bien variaciones más complejas como cambios de perspectiva y escala. Con el término ‘cambio de escala’ nos referimos a variaciones que se puedan dar en el tamaño de un objeto que aparezca en la imagen, debido por ejemplo, a un desplazamiento longitudinal de la cámara.

Para realizar la búsqueda de este conjunto de puntos de interés existen diferentes métodos. En función del tipo que busquemos utilizaremos una técnica u otra. Es decir, ciertos operadores aplicados sobre la imagen aportan información sobre píxeles con carácter de esquina, mientras que otros pueden dar información sobre bordes. Es importante definir previamente qué tipo de puntos se quieren detectar, ya que algunos serán más estables a determinadas transformaciones, mientras que otros serán invariantes a otro tipo de cambio. Por lo tanto, una vez desarrollado nuestro software, esperaremos que sea robusto a las transformaciones que hayamos definido previamente en la elección del tipo de puntos característicos.

El siguiente paso, una vez calculado dónde están las zonas de interés, es describir la zona en cuestión. Los puntos concretos que nos proporcionan cualquiera de los métodos anteriores nos aportan información de localización, sin embargo, no es suficiente para una posterior búsqueda de correspondencias entre imágenes. Por ese motivo, una vez los métodos anteriores nos dicen dónde se puede extraer información, hay que describir el vecindario del punto. Dicha descripción de los alrededores del punto se llevará a cabo en un radio determinado. Cuanto mayor sea este radio, mayor coste computacional conllevará su cálculo, pero se describirá con mayor amplitud cada región de la imagen. Por el contrario, no interesan descriptores demasiado grandes, ya que en ese caso se perdería el concepto de ‘localidad’ en los descriptores. Por tanto, hay que llegar a un término medio que proporcione un algoritmo eficiente y a su vez que describa suficientemente cada zona importante.

Existen diferentes maneras de describir localmente los alrededores del punto característico. Una forma puede ser evaluando el nivel de gris de los píxeles vecinos. Partiendo de que la imagen está normalizada a la unidad, aquellos píxeles más oscuros tendrán un valor cercano a cero y los más claros serán próximos a uno. También cabe la posibilidad de evaluar el nivel de color. Para ello, hay que tener en cuenta que hay que analizar las tres matrices que definen el color: R, G y B. El color

de una región específica puede ser representado mediante sus tres histogramas de color, o bien calculando la media de la región (y por tanto obteniendo tres escalares).

Dejando de lado los niveles tanto de gris como de color, una tercera opción es la de calcular y detallar las orientaciones de los gradientes alrededor del punto característico. De la misma forma que antes, se pueden estudiar mediante el uso de histogramas o el cálculo de medias, entre otros. Esta multitud de posibilidades en el momento de la descripción local de la imagen es independiente del método escogido a la hora de buscar los puntos de interés.

La descripción de los vecindarios nos permitirá asociar los puntos clave de una imagen de entrenamiento con otra de test, y por tanto seremos capaces de identificar puntos correspondientes entre ellas. Por ese motivo es tan importante que todos estos descriptores locales sean insensibles a cambios de escala, rotaciones, traslaciones, etc. Mientras en una imagen A (de entrenamiento) aparece un objeto en una posición determinada, en una imagen B (de test) puede aparecer el mismo objeto rotado o escalado de diferente forma. Sin embargo, el algoritmo debe ser invariante a esos cambios, gracias a las propiedades singulares de los puntos característicos y sus correspondientes descriptores.

Históricamente, han existido multitud de técnicas que nos permiten buscar puntos de interés. Inicialmente, la mayor parte de ellas se basaban en la detección de esquinas o *corners*, ya que éstas son especialmente robustas frente a cambios en escala, rotación, orientación, iluminación y otros factores. El detector de esquinas propuesto por *Moravec* [7] fue una de las primeras implementaciones (70's), aunque más tarde fue mejorada por *Harris* [8] (1988).

El detector de esquinas de Harris fue también mejorado por *Jianbo Shi* y *Carlo Tomasi* [9] en 1996, y es uno de los algoritmos más utilizados en la detección de puntos característicos. Actualmente se usa más que Harris debido a que detecta esquinas que Harris no siempre detecta y es algo más resistente a la rotación. También existen detectores de bordes o *edges*, como *Canny* [10] (1986), aunque éstos tienden a dar más problemas ante cambios de perspectiva que los anteriores. Posteriormente, aparecieron teorías más complejas que trabajaban con diferentes escalas de una imagen para buscar puntos característicos (*scale-space theory* [11], 90's), a partir de la parametrización del tamaño del filtro con la que son filtradas. Un ejemplo es el método SIFT ([12] y [13]) (Sc*ale-Invariant Feature Transform*) publicado y patentado por David Lowe. La mejora de este algoritmo es conocida como SURF [14]. Su principal ventaja respecto a SIFT es la rapidez de ejecución.

Ante la multitud de opciones, en este trabajo se ha optado inicialmente por implementar el detector de esquinas propuesto por Shi y Tomasi, debido a su uso extendido en la comunidad de la visión por computador.

Después de analizar los resultados obtenidos, concluimos en que puede resultar de utilidad ante cambios simples (traslaciones o rotaciones), pero insuficiente para variaciones más complejas (cambios de luminosidad, de perspectiva y escala). Por ello, se consideró necesario utilizar una implementación de un método más robusto como SURF.

Este método está basado en la teoría de *scale-space* comentada anteriormente. A la vez que es mucho más complejo computacionalmente, a priori

proporcionará mejores prestaciones. Después de un proceso complejo, podremos construir los descriptores para poder calcular correspondencias entre diferentes fotografías.

2.4. OBTENCIÓN DE MAPAS DE DISPARIDAD

Dada la posición de los ojos en los humanos y la forma de moverlos las imágenes que se reciben en cada ojo son prácticamente iguales, con una diferencia en la posición relativa de los objetos. Estas diferencias relativas en la posición en cada imagen (la disparidad), tiene una relación directa con la distancia (profundidad) a la que se encuentran los objetos entre sí y respecto del observador. El cerebro es capaz de interpretar esa diferencia y reconstruir la estructura de la escena que ve el observador. Según Marr y Poggio [16] existen tres etapas en el proceso de recuperación de la estructura de una escena.

Estas son, primero, seleccionar un punto característico de un objeto en una de las imágenes (vistas por cada ojo), segundo, encontrar el mismo punto característico en la otra imagen, y tercero, medir la diferencia relativa (disparidad) entre la posición de estos dos puntos.

En las últimas tres (casi cuatro) décadas el tema de la visión estéreo ha sido abordado por la comunidad de *Computer Vision*. Se llama visión estéreo a la capacidad de recuperar la estructura tridimensional de una escena a partir de, por lo menos, dos vistas o imágenes diferentes de la misma. La estructura que se recupera es la posición de los objetos presentes en la escena, fundamentalmente recuperando la profundidad (distancia al observador) de los objetos. Una formulación alternativa de este problema puede ser la de localizar para cada punto de cada una de las imágenes su correspondiente en la otra imagen; entendiendo por puntos correspondiente aquellos que son proyecciones del mismo punto del espacio en cada una de las imágenes

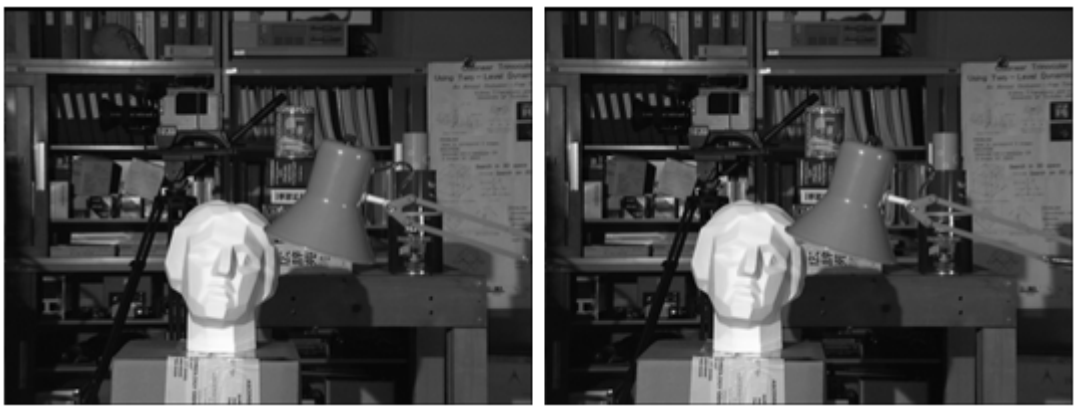

FIGURA 8 : IMÁGENES ESTÉREO. IMÁGENES IZQUIERDA Y DERECHA DE LA MISMA ESCENA CON UN DESPLAZAMIENTO HORIZONTAL (HACIA LA DERECHA) DE LA CÁMARA.

De esta forma se genera una *imagen* densa de profundidades para cada uno de los puntos de la escena proyectados en ambas imágenes La Figura 8 muestra un par de imágenes de una escena tomadas con un pequeño desplazamiento horizontal como serán vistas por el ojo izquierdo y derecho, respectivamente. A continuación se muestra el mapa de profundidades real del par de imágenes anteriores, referente a la imagen izquierda.

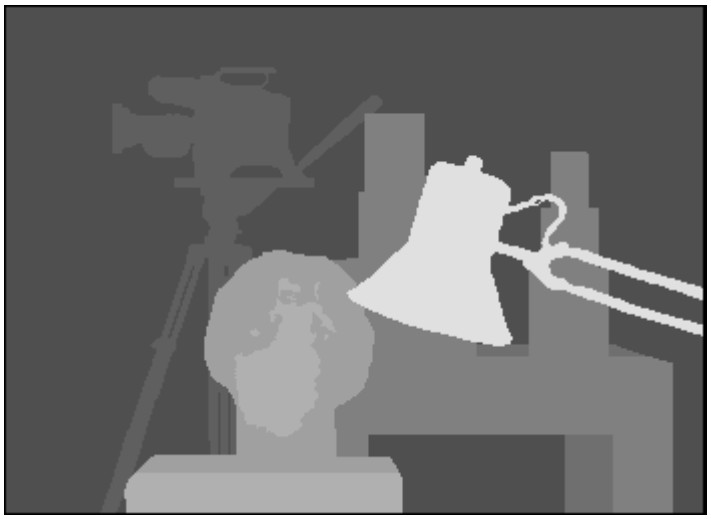

FIGURA 9 : MAPA DE PROFUNDIDAD RELATIVO A LA IMAGEN IZQUIERDA DE LA FIGURA 8

Lograr que una computadora *pueda ver* es un desafío en la comunidad de *Computer Vision* desde sus principios y que aún no ha sido resuelto completamente, mas allá de lo que se entienda por *pueda ver* en el caso de una computadora. Existen varias dificultades que se plantean cuando se intenta abordar este problema utilizando una computadora como herramienta de procesamiento; por ejemplo, la adquisición de las imágenes y la preparación para su tratamiento, el ruido en la adquisición, diferencias de intensidad/color del mismo punto del espacio en las dos imágenes, las oclusiones y complejidad de la escena, etc.

Sin embargo muchos avances se han realizado logrando resultados importantes en diversas aplicaciones. Una de las aplicaciones de una imagen densa de profundidades de la escena es la descomposición de una imagen en capas de igual profundidad para su posterior procesamiento y generación de nuevas vistas (*Image Based Rendering*). Se utiliza en la reconstrucción tridimensional de un objeto a partir de varias vistas o una secuencia de video. También en la navegación de robots, creación de realidad virtual, codificación de imágenes estéreo seguimiento y vigilancia (conteo de personas), etc.

En los animales la capacidad de recuperar la estructura tridimensional está dada por la existencia de dos sensores, normalmente los ojos, aunque también pueden ser los oídos como el caso de los búhos y murciélagos. A partir de las *imágenes* obtenidas por cada uno de los sensores, entre los cuales debe existir una distancia espacial, es posible recuperar la geometría tridimensional.

Esto no es igual en todos los animales, a pesar de tener dos ojos. Animales con una visión lateral, debida a tener los ojos a los lados del cuerpo, no tienen la

misma capacidad para recuperar la estructura tridimensional, que los animales con ambos ojos al frente. Normalmente los primeros, son animales que deben protegerse de los predadores, que son los segundos. Para un predador es fundamental poder medir exactamente la distancia a una presa para poder hacer el ataque justo; mientras que para los otros es necesario mantener una vigilancia periférica para poder detectar movimientos que puedan significar peligro. Por esto es importante que los predadores tengan un buen sistema binocular estéreo, preparado para tareas específicas con el fin de la supervivencia. Los animales con visión periférica igualmente logran recuperar la estructura tridimensional que los rodea, pero sin la precisión que logran los animales con visión frontal. Por otro lado, los animales con visión frontal obtienen una visión periférica a partir de la visión frontal con movimientos de la cabeza, con un cuello mas desarrollado.

En los humanos, la percepción visual de la estructura tridimensional es realizada por el Sistema Visual Humano (SVH) con los ojos y el cerebro. El aprendizaje (o adaptación) permite poder realizar tareas sencillas a pesar de tener anuladas o debilitadas las capacidades de visión estéreo. Por ejemplo, agarrar una taza con un solo ojo abierto. La precisión obtenida no es la misma pero igual se pueden realizar tareas básicas. Igualmente, es posible engañar la percepción que se obtiene de algunas imágenes, cuando el cerebro intenta recuperar una estructura tridimensional a partir de un dibujo plano.

Un ejemplo sencillo se puede ver en la Figura 10.

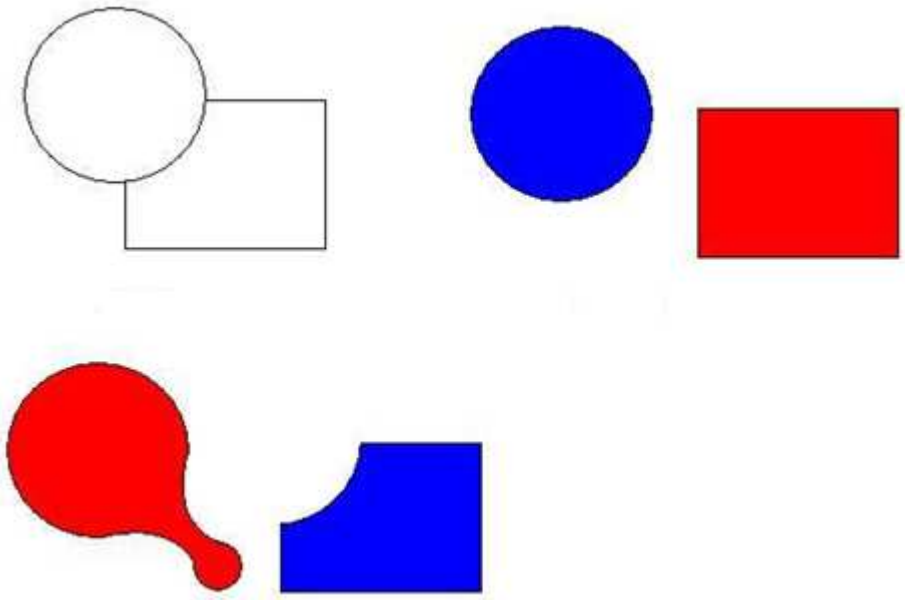

FIGURA 10 : DIFERENCIA DE PERCEPCIÓN

Observando inicialmente la imagen superior izquierda podemos creer que el circulo esta en primer plano respecto al rectángulo. La imagen superior derecha muestra estos dos objetos por separado. Pero con esa sola imagen no se puede asegurar que la figura que está en primer plano sea un círculo. Se puede dar el caso de la imagen inferior por ejemplo en el que la figura azul está en primer plano respecto de la roja.

Podemos afirmar que conociendo la disparidad entre dos puntos somos capaces de calcular la profundidad a la que se encuentra el punto real en el espacio tridimensional. Por ello la tarea de hallar la correspondencia es vital.

2.4.1. DEFINICIÓN DE DISPARIDAD

Una forma de estimar la profundidad de cada uno de los puntos en la escena es mediante el cálculo de la disparidad entre las imágenes de la misma. Asumiremos que la escena es estática, es decir, los objetos visibles en la escena no cambian su posición en la misma ni sufren deformaciones.

Para definir la disparidad asumamos una configuración de dos cámaras de características similares, como la que se muestra en la Figura 11.

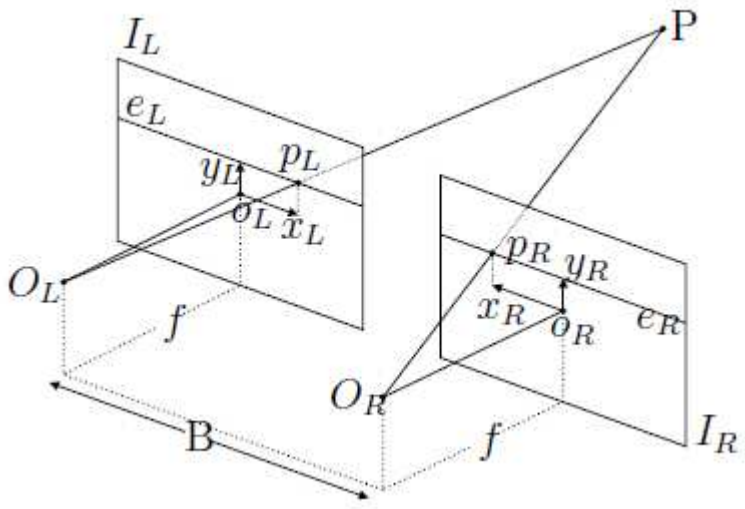

FIGURA 11 : ESQUEMA DE CONFIGURACIÓN DE DOS CÁMARAS SIMPLES

Estas dos cámaras forman un par estéreo, y asumiremos que cada una de ellas cumplen un modelo *pinhole*. Los ejes ópticos de las cámaras son paralelos, ORoR || OLoL. Ambas cámaras tienen la misma distancia focal, f, con centros OL y OR separados una distancia B, llamada línea base (*baseline*), de forma que las imágenes que se forman, IL e IR, estén en planos paralelos. De esta manera la línea base es paralela a la coordenada x de las imágenes. Con el modelo *pinhole* considerado, un punto en el espacio tridimensional P, con coordenadas (X; Y;Z), se proyecta en cada una de las imágenes bidimensionales en los puntos pL y pR, con coordenadas (xL; yL) y (xR; yR), respectivamente.

El plano que contiene a los puntos P, OL y OR, interseca a las imágenes en dos rectas eL y eR que se denominan líneas epipolares. Un punto, pL, en la recta eL de la imagen IL tiene su correspondiente en algún punto de la recta eR. Esto reduce la

búsqueda del correspondiente de pL de toda la imagen IR a la recta eR. Dada la configuración específica del par estéreo, las líneas epipolares de ambas imágenes son horizontales y están alineadas, lo que facilita aún más la búsqueda de puntos homólogos. En la siguiente figura podemos ver como se relacionan los parámetros definidos en el par estéreo, que permiten obtener la relación entre la disparidad d y la profundidad Z del punto P.

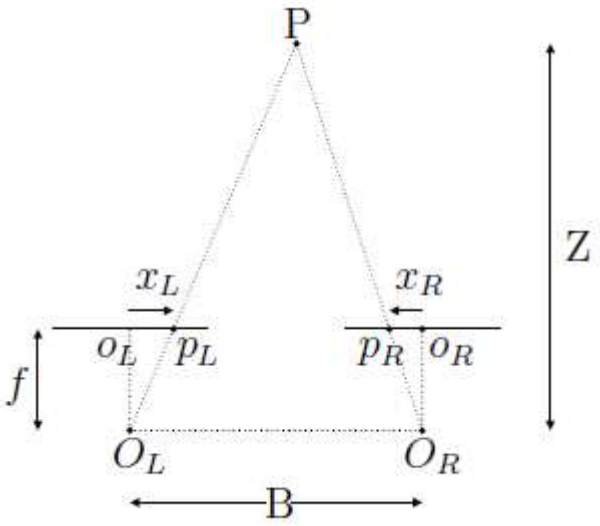

FIGURA 12 : ESQUEMA DE DISPARIDAD

La disparidad es la diferencia en las coordenadas horizontales de los puntos pL y pR, o sea, d = xL-xR. Dependiendo el sistema de referencia utilizado en las imágenes, la definición puede cambiar de forma que el signo sea siempre positivo. Las coordenadas de pL y pR quedan relacionadas mediante:

$$\begin{cases} x_L = x_R + d \\ y_L = y_R \end{cases}$$ Ec. 10

Considerando los triángulos POLOR, pRoROR y pLoLOL, utilizando semejanza entre triángulos se llega a:

$$d = \frac{f}{Z} B$$ Ec. 11

Entonces, se tiene la relación entre d y Z:

$$d \propto \frac{1}{Z}$$ Ec. 12

Basados en la última ecuación podemos recuperar, salvo por una constante de escala, la profundidad de cada píxel en cada una de las imágenes a partir de la disparidad calculada.

La relación de proporcionalidad inversa planteada en la ecuación 12 es fácilmente verificable observando alternadamente las imágenes izquierda y derecha, y notando que los objetos más cercanos a la cámara (menor Z) tienen mayor desplazamiento relativo (mayor d) en las imágenes.

Los algoritmos de cálculo de disparidad obtienen una imagen con el valor de disparidad calculado en cada punto de las imágenes de entrada. Estas imágenes se conocen como mapas de disparidad; en la Figura 9 pudimos observar el mapa de disparidad real de las imágenes de la Figura 8 para la imagen de la izquierda.

Algunos puntos de la escena son visibles en una sola de las imágenes de entrada, o sea se proyectan en una sola de las imágenes. En estos puntos se producen oclusiones por la disposición de los objetos en la escena y la disparidad no puede ser calculada por este método. La proyección con el modelo pinhole considerado, que lleva las coordenadas del punto en el espacio P = (X; Y;Z) al punto del plano p = (x; y), no es una transformación lineal. Para obtener una transformación lineal cerrada se hace una extensión de la geometría euclídea a la geometría proyectiva, introduciendo las coordenadas homogéneas, añadiendo una nueva coordenada 1. La representación en coordenadas homogéneas para P y p quedan:

Ph = (X; Y;Z; 1)

ph = (x; y; 1)

Con esta nueva representación las propiedades del punto (kx; ky; k) para cualquier k ~= 0 son iguales, y todas se considera representantes del punto p = (x; y).

Con el agregado de esta nueva coordenada es posible definir un mapeo lineal entre las coordenadas homogéneas de un punto del espacio Ph y las coordenadas homogéneas del punto de la imagen ph su proyección mediante el modelo de la cámara. Esta transformación lineal puede representarse mediante un producto matricial con la *matriz de la cámara* PC [1]:

ph = PC * Ph

La forma que toma PC varía dependiendo el tipo transformación que se considere (afín, proyectiva, etc.). La forma más sencilla que toma en el caso del modelo de cámara considerado es:

$$P_c = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$ Ec. 13

Donde

$$\alpha_u = f k_u$$ Ec. 14

$$\alpha_v = f k_v$$ Ec. 15

$$(u_0, v_0)$$ Ec. 16

son los parámetros intrínsecos de la cámara. Escribiendo la ecuación 13, con ku = kv = 1 y (u0; v0) = (0; 0) se obtiene la relación entre las coordenadas horizontales y verticales en la imagen a partir de las coordenadas del punto en el espacio (X; Y;Z) y la distancia focal de la cámara (en píxeles):

$$\begin{pmatrix} kx \\ ky \\ k \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Ec. 17

Junto con la ecuación 12 dan las ecuaciones para recuperar la estructura de la escena, a menos de una constante,

$$\begin{cases} x = \frac{f}{Z} X \\ y = \frac{f}{Z} Y \\ d = \frac{f}{Z} B \end{cases}$$

Ec. 18

Estas ecuaciones pueden verificarse geométricamente en la Figura 12. Si se conocen los parámetros intrínsecos de la cámara y la configuración geométrica del par estéreo (f y B en el caso más simple) es posible recuperar la profundidad de cada punto de la imagen, y por lo tanto la estructura euclídea de la escena.

2.4.1.1. TÉCNICAS DE CORRESPONDENCIA UTILIZADAS

El análisis de la visión en estéreo tiene una historia relativamente corta. Los primeros artículos específicos que se encuentran en la literatura datan de los años 70. Desde entonces han surgido multitud de ideas para resolver el problema de la correspondencia que es el más esquivo, y quizá por ello el más importante. Actualmente se siguen buscando soluciones a este problema, ya que no se ha encontrado una solución que funcione bien con imágenes sintéticas y con imágenes reales, en ausencia y en presencia de ruido, etc. Debido a esa multitud de intentos de resolución que han aparecido hasta la fecha y a que todavía se siguen buscando nuevos métodos, el intentar abarcar absolutamente todos los intentos sería un trabajo inacabable. Por ello, en este apartado, se van a repasar algunos de los intentos de solución más representativos. Se van a intentar presentar las tres principales vertientes con el fin de centrar la atención en los conceptos que se presentan en cada modalidad.

2.4.1.2. TÉCNICAS BASADAS EN LA CORRELACIÓN

Las técnicas de área basadas en intensidad han sido investigadas extensamente para aplicaciones comerciales en estéreo fotogrametría. La principal de

este tipo de técnicas es la técnica de correlación de área. Ésta se basa en considerar los valores de intensidad de los píxeles de las imágenes como una señal bidimensional, que en una de las dos imágenes ha sufrido una traslación, lo que nos lleva al concepto de disparidad. Se trata de obtener, para cada punto de la imagen, dicha traslación, minimizando una función de coste, que comúnmente tiene que ver con la correlación. Para cada píxel de una imagen se calcula la correlación entre la distribución de intensidades de una ventana centrada en dicho píxel y una ventana del mismo tamaño centrada en el píxel a corresponder de la otra imagen. Esta técnica aplica, además de la restricción epipolar (es decir, que el punto homólogo esté en la línea epipolar), las restricciones Lambertiana (superficies difusas), de continuidad y otra restricción conocida como frontoparalela, que asume que la disparidad es constante localmente, por lo que las superficies deben ser paralelas a los planos de imagen de las cámaras, o al menos tener una pendiente pequeña. Las ventajas de utilizar este método de correlación de área, es que se obtienen unos buenos resultados en imágenes con texturas importantes y son fáciles de paralelizar. Además, permite crear mapas densos de disparidad, es decir, se obtendrá una disparidad para todos los puntos de la escena, y no solo para los contornos, esquinas u otras primitivas de mayor nivel. También es cierto que presenta problemas con imágenes que contienen elevadas discontinuidades de superficie y es una técnica muy sensible a variaciones fotométricas debidas a sombras o reflejos. Tiene además problemas con las oclusiones y requiere de un proceso posterior de eliminación de falsas correspondencias. También es posible utilizar esta técnica como complemento de otras e incluso realizar algún tipo de postprocesado sobre el mapa de disparidad hallado, que permita reducir los inconvenientes de la correlación de área como técnica de correspondencia.

2.4.1.3. TÉCNICAS DE RELAJACIÓN

La técnica de correlación de área por sí sola presenta numerosos errores en las correspondencias, que bien pueden ser eliminados mediante un postprocesado o parcialmente evitados mediante un proceso que se conoce con el nombre de relajación o algoritmo cooperativo. La idea básica de las técnicas de relajación es permitir a los píxeles que se van a poner en correspondencia, realizar “estimaciones controladas” de cómo debe ser su correspondencia y, después, permite a las correspondencias reorganizarse propagando algunas de las restricciones descritas en los apartados anteriores. Para este tipo de proceso, no solamente importa el valor de la correlación obtenida para los píxeles de la línea que se analiza, sino que también otorga importancia a los valores de correlación obtenidos para una cierta vecindad que se conocerá con el nombre de región excitatoria o de soporte, y un grupo de píxeles que se conocerán como región inhibitoria. Este algoritmo se implementa a partir del denominado cubo de correlación, que será una matriz de tres dimensiones (filas x columnas x disparidad). Cada uno de los elementos de la matriz almacenará el valor de la correlación obtenido para la ventana de la imagen 1 centrada en el píxel marcado por las coordenadas (fila, columna) con la ventana de la imagen 2 centrada en el píxel de coordenadas (fila, columna + disparidad). Este cubo de correlación se muestra en la Figura 13.

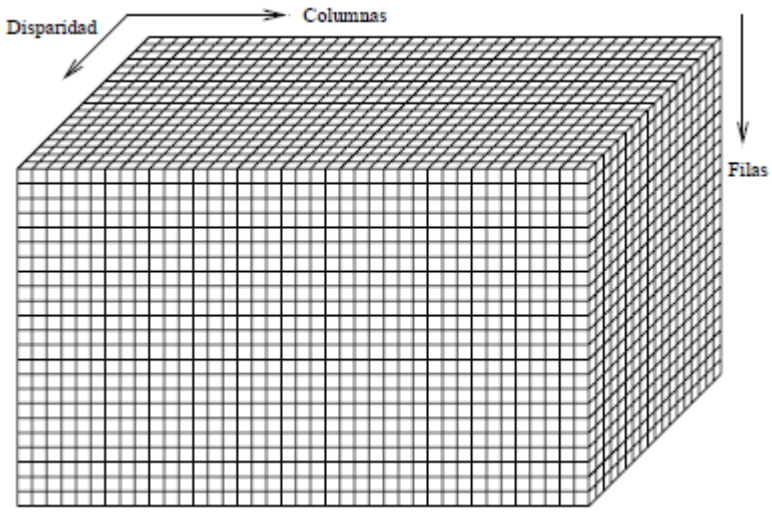

FIGURA 13 : CUBO DE CORRELACIÓN

Una vez creado el cubo de correlación para cada uno de los píxeles de la imagen marcado por su fila y su columna, se tiene un vector unidimensional de tamaño el límite de disparidad elegido en el análisis, y que almacena los valores de la correlación.

El proceso de relajación se realizará ahora línea por línea, o fila por fila, de modo que, para cada una de las filas, se tiene una matriz bidimensional de la magnitud horizontal y la disparidad. Es en esta matriz donde se definirán las que antes se han llamado región inhibitoria y excitatoria.

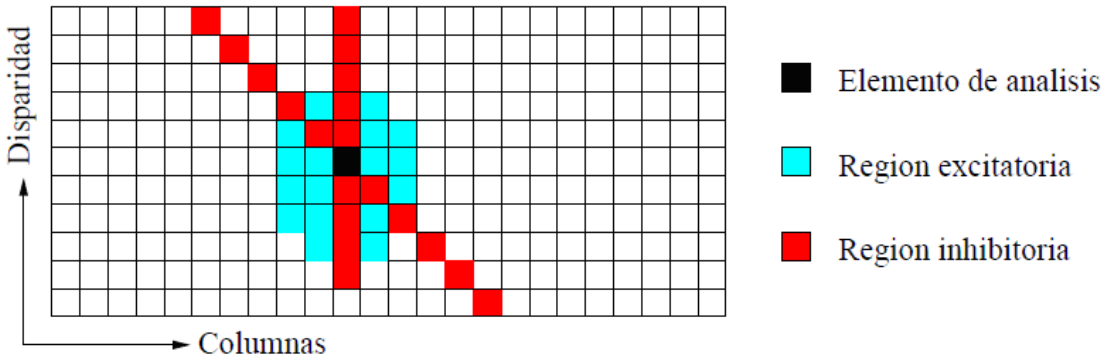

FIGURA 14 : REGIÓN EXCITATORIA E INHIBITORIA

Las regiones excitatoria e inhibitoria se definen utilizando las restricciones de continuidad, unicidad y, de forma indirecta, también de ordenamiento, además de la restricción epipolar que se ha utilizado para crear el cubo de correlación. Para cada píxel de coordenadas (*ul, vl*), se tratará de buscar un valor de disparidad *d* que se

corresponda con un píxel en la imagen D. Si ese punto es realmente una correspondencia tendrá un alto valor de correlación y, a su vez, atendiendo a la restricción de continuidad, los puntos cercanos a él también tendrán valores altos, de modo que esos puntos cercanos serán la región excitatoria. Por el contrario, si dicho punto es correcto, los demás elementos de la matriz cuya coordenada *X* sea la misma pero difieran en la disparidad, serán correspondencias falsas, de modo que tendrán valores pequeños de disparidad. Si a su vez se aplica la restricción de ordenamiento y unicidad, los puntos que tengan una coordenada *X* diferente pero tengan una disparidad tal que lleven al mismo píxel, no serán correspondencias válidas, por lo que sus valores de correlación también serán pequeños. Estos dos últimos grupos de puntos se corresponderán con la región inhibitoria. En la Figura 14 se muestra gráficamente todo esto. Conociendo las características de las regiones comentadas, será posible utilizarlas para mejorar el proceso de la búsqueda de las correspondencias.

2.4.1.4. TÉCNICAS DE PROGRAMACIÓN DINÁMICA

Este método trata el problema de la correspondencia entre primitivas. Una correspondencia entre primitivas puede ser, para hacernos una idea, una correspondencia entre contornos de las dos imágenes. Así, este problema entre imágenes puede ser abordado como minimización de una función de coste. La programación dinámica es una forma eficiente de minimizar (o maximizar) funciones de gran número de variables discretas. Un intento satisfactorio utilizando programación dinámica para resolver el problema de la correspondencia estéreo es el expuesto por Ohta y Kanade [18] que utiliza los contornos como primitivas básicas.

Asumamos que las líneas epipolares son paralelas a las filas de las imágenes y consideremos dos líneas correspondientes en las imágenes derecha e izquierda, llamadas *scan-lines*. En cada fila se identifican varios píxeles de contorno, y se incluyen los dos finales de las líneas por conveniencia. La correspondencia de estos píxeles de contorno puede considerarse como el problema de corresponder los intervalos entre ellos de la siguiente manera: Ordenamos los píxeles de contorno de izquierda a derecha en cada línea y los numeramos entre 0 y *N*-1 en la imagen izquierda y 0 a *M*-1 en la derecha. En la Figura 15 se representan los pares (*i,j*) de puntos de contorno de las líneas derecha e izquierda como puntos que forman una rejilla. Corresponder el intervalo [*i,i'*] de la izquierda con el [*j,j'*] de la derecha es equivalente a dibujar un segmento entre los puntos *m*=(*i,j*) y *m'*=(*i',j'*) en la rejilla. El objetivo es encontrar una secuencia de segmentos (un camino) desde el punto *m*0=(0,0) hasta el punto *me*(*N*-1, *M*-1). En esta búsqueda podemos aplicar restricciones del estilo de las vistas en apartados anteriores. La restricción de orden es interesante ya que es equivalente a decir que los caminos admisibles son caminos monótonos decrecientes.

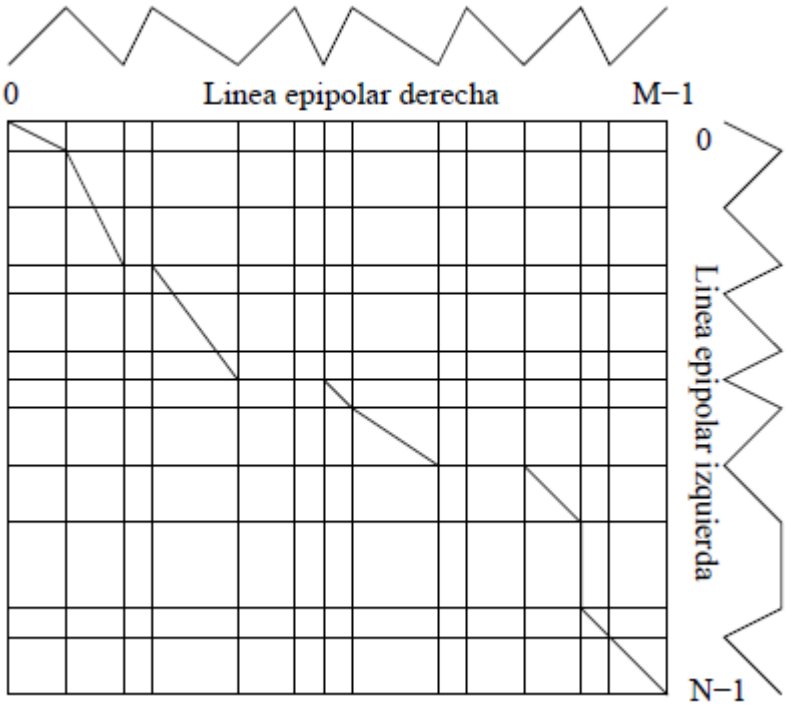

FIGURA 15 : REJILLA DE CORRESPONDENCIA (PROGRAMACIÓN DINÁMICA)

A pesar de esta restricción todavía existen muchos caminos posibles. Definiremos el mejor camino como el que minimice una función de coste. Primero definimos el coste $c(m,m')$ de un segmento entre los puntos $m=(i,j)$ y $m'=(i',j')$. Por supuesto, existen varias formas de definir esta función de coste. En general esta función de coste debe medir dos cosas: (1) la similitud de las características de los píxeles de la imagen derecha y la izquierda, características que pueden ser, por ejemplo, la orientación de los contornos o contraste a través de ellos, y (2) la similitud de intensidades a lo largo de los intervalos entre contornos. Si existe un contorno entre dos filas, las correspondencias de una fila deben depender bastante de las vecinas. Reforzar la consistencia es equivalente a aplicar la continuidad de las figuras, y hay varias formas de hacerlo: (1) utilizando un proceso cooperativo para detectar y corregir los resultados de la correspondencia, mientras que (2) Ohta y Kanade [18] lo incluyen en la función de coste y resuelven una programación dinámica en un espacio 3D en vez de un espacio previo 2D. Además de la minimización de la función de coste para las líneas epipolares, también se pueden aplicar relaciones de correspondencia entre líneas epipolares vecinas (superiores e inferiores) con el fin de reducir la ambigüedad. Algunos ejemplos que avanzan en esta dirección son los llamados “graph cuts”, que tratan de minimizar una función de coste que implica tanto a la dirección horizontal como a la vertical. La principal desventaja de la programación dinámica es la probabilidad de que errores locales se puedan propagar a lo largo de la línea epipolar descartándose correspondencias potencialmente correctas.

2.4.1.5. CORTE DE GRAFOS

Una extensión del concepto de buscar la correspondencia entre *scanlines* correspondientes independientemente de otras *scanlines*, es la de buscar la correspondencia de todas las *scanlines* simultáneamente. Este nuevo concepto logra pasar de buscar una correspondencia entre *scanlines* considerándolas independientes entre sí, a buscar una correspondencia de una superficie de mínimo costo. Ajustar una superficie minimizando alguna energía, presenta una coherencia local en todas las direcciones, en particular la horizontal (*intra-scanline*) y la vertical (*inter-scanline*), por la forma en que se construye la solución, y no se fuerza esta coherencia mediante restricciones en la minimización. La utilidad del corte de grafos viene dada por la capacidad para minimizar una cierta energía. Dependiendo de la expresión que se plantee los resultados pueden ser aplicables a varios problemas. Boykov y Kolmogorov [19,20] presentan una comparación de varios algoritmos utilizando variantes de una expresión de energía en aplicaciones como restauración de imágenes, calculo de disparidad, y segmentación interactiva de imágenes. Kolmogorov y Zabih [21] caracterizan las funciones de energía que pueden ser minimizadas con este tipo de formulaciones, planteando condiciones necesarias y suficientes; y presentan una construcción general del grafo para la minimización.

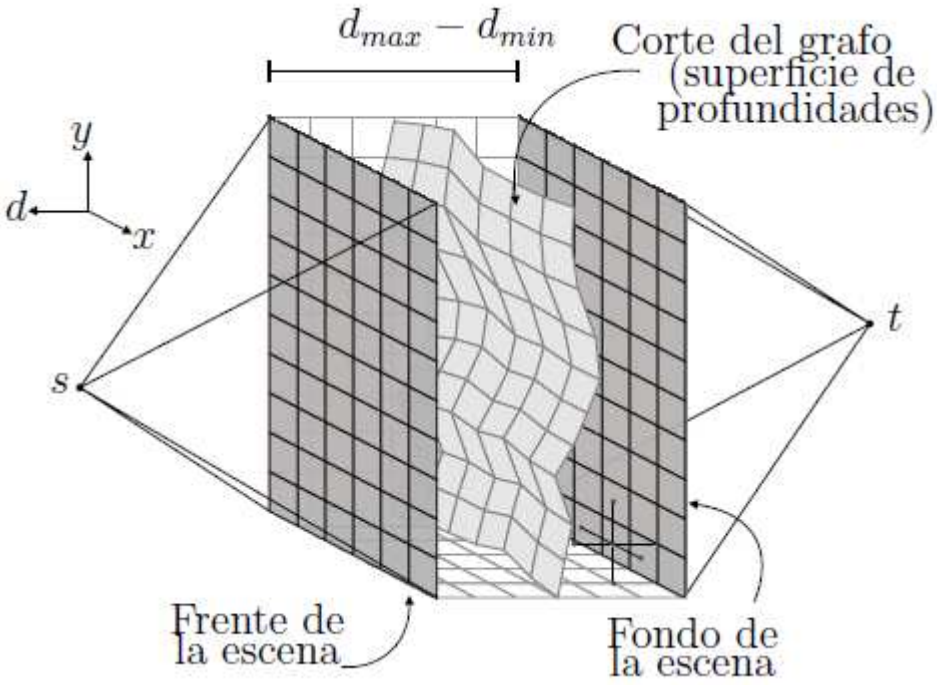

FIGURA 16 : REPRESENTACIÓN DEL PROBLEMA DE CÁLCULO DE DISPARIDAD MEDIANTE CORTE DE GRAFOS.

En la Figura 16 podemos ver el problema de cálculo de disparidad mediante corte de grafos. El grafo se arma de forma que cada nodo (x; y; d) del mismo está

conectado con cuatro puntos a igual disparidad d (dos en la horizontal y dos en la vertical), y con dos a disparidades d - 1 y d + 1. La superficie representa el corte del grafo que minimiza alguna expresión de energía.

A continuación se presenta un repaso de los principales algoritmos que hacen uso del corte de grafos, que por su importancia y aplicación al cálculo de disparidad sobresalen. La diferencia entre los distintos algoritmos y las aplicaciones se da en la forma en que se construye el grafo y la expresión de la energía a minimizar.

El método de corte de grafos es anterior a los problemas de visión, la primera implementación en el área de estéreo fue la realizada por Roy y Cox en 1998. Roy y Cox [22] presentan un algoritmo para solucionar el problema de correspondencia con N cámaras utilizando corte de grafos. A partir de N imágenes de una escena se recupera el mapa de disparidad para una de las vistas, con las triangulaciones de las N - 1 imágenes restantes. El grafo construido tiene un nodo por cada píxel de la imagen en cada posible valor de disparidad, o sea (dmax - dmin) * m * n, mas los dos nodos terminales, s y t, donde m y n son el numero de filas y de columnas de las imágenes. Con el agregado de los nodos terminales se tiene una estructura de grafo. La asignación de las capacidades a los enlaces es de acuerdo a la siguiente convención:

Los enlaces entre nodos en la dirección de los ejes de las imágenes → coclu().

Los enlaces entre nodos en la dirección de la disparidad → cdisp() = k*coclu().

El parámetro k controla la suavidad de la solución obtenida.

La forma de resolver el corte del grafo es con uno de los algoritmos clásicos [24] conocido como preow-push lift-to-front.

Boykov y otros [21] presentan un nuevo algoritmo para hallar el corte de grafos que encuentra un mínimo local en forma más eficiente (más rápida) que los algoritmos tradicionales [19,20]. Este algoritmo presenta dos variantes (alfha-expansiones y alpha-betas-waps) y han sido utilizados en los últimos algoritmos de cálculo de disparidad que presentan los mejores resultados [19, 20]. La minimización se plantea de la siguiente forma: encontrar una configuración de etiquetas L = {Lp/p en el grafo}, que minimizan una energía de la forma:

$$E(L) = E_{data}(L) + E_{smooth}(L) \qquad \text{Ec. 19}$$

donde Esmooth(L) mide la suavidad que presenta la solución, y Edata(L) mide la diferencia entre L y los datos originales. Para el caso de cálculo de disparidad, las etiquetas que se desean asignar son los posibles valores de disparidad. Un caso particular de la energía es la energía de Potts:

$$E_p(L) = \sum_{p \in P} D_p(L_p) + \sum_{\{p,q\} \in N} D_{p,q} T(L_p \neq L_q) \qquad \text{Ec. 20}$$

donde Dp(Lp) es un costo por tener una disparidad Lp (la etiqueta para el punto p) y Dp,q es un potencial de penalización cuando una pareja de puntos {p,q} en un vecindario N tiene diferentes etiquetas (disparidades), T( ) vale 1 si el argumento es verdadero o 0 si es falso.

Kolmogorov y Zabih [21] presentan un algoritmo donde plantean una formulación de la energía de Potts en la cual contemplan las oclusiones, y lo resuelven con el algoritmo presentado por Boykov y otros [23]. La energía para una configuración L, que plantean, tiene la expresión:

$$E(L) = E_{data}(L) + E_{smooth}(L) + E_{oclu}(L) \qquad \text{Ec. 21}$$

Este algoritmo es el que está dando los mejores resultados prácticos según los resultados que presentan los autores, y por otros trabajos de comparación de los distintos métodos existentes para la resolución del cálculo de disparidad.

Los mismos autores presentan [25] una variante de este algoritmo para la reconstrucción de una escena a partir de N imágenes, variando la expresión de la energía en la ecuación 21.

2.4.2. OCLUSIONES

La mayor parte de la investigación en estéreo visión en la última década se ha orientado a la detección y medida de regiones ocultas en una de las imágenes y en recuperar la profundidad precisa para estas regiones. Este apartado define el problema de la oclusión en estéreo visión y contempla tres clases de algoritmos para el manejo de oclusiones:

- los métodos que detectan oclusiones
- los métodos que reducen la sensibilidad a las oclusiones
- los métodos que modelan la geometría de las oclusiones.

El problema de las oclusiones en estéreo visión se refiere al hecho de que algunos puntos de la escena son visibles por una cámara y, sin embargo, no por la otra, debido a la propia escena y a la geometría del sistema. En estos casos, la estimación de la profundidad no es posible si no se añaden más vistas en las que el punto no esté oculto, o si no se asumen ciertas características de la escena.

2.4.2.1. MÉTODOS DE DETECCIÓN DE OCLUSIONES

Los acercamientos más simples al manejo de oclusiones comienzan por su detección previa, o posterior a las correspondencias. Estas regiones en algunos casos resultan interpoladas cuando se pretende conseguir un mapa denso de disparidades, o simplemente no se toman en consideración cuando se busca un mapa menos denso. La aproximación más común es detectar discontinuidades en el mapa de profundidad después del análisis de correspondencias. Habitualmente se utilizan filtros de mediana

para eliminar dichas discontinuidades que son producidas generalmente por oclusiones. Las disparidades inconsistentes se consideran producidas por oclusiones en la escena. Existen otras muchas causas posibles de inconsistencia, incluyendo diferencias de perspectiva, iluminación no uniforme o ruido en los sensores. La inconsistencia de izquierda a derecha trata todos estos fenómenos por igual, pero es un método que al utilizar las funciones SAD o SSD tiene un coste computacional razonable. Por ello, estas funciones son comúnmente utilizadas en sistemas en tiempo real. La restricción de ordenamiento también se puede utilizar para detectar oclusiones. El ordenamiento relativo de los puntos a lo largo de las líneas epipolares es monótono, asumiendo que no existen objetos excesivamente estrechos en la escena. Otra aproximación a la detección de oclusiones se basa en la observación de discontinuidades en la profundidad y orientación que aparecen en torno a los bordes de los objetos. Los mapas de disparidad se suavizan, manteniendo exclusivamente sin suavizar las disparidades asociadas a los bordes. Entonces, aquellos puntos con grandes diferencias de disparidad entre la versión original y la suavizada se consideran como regiones ocultas. La programación dinámica de Ohta y Kanade [18] que hace corresponder a las regiones a través de la interpolación de las profundidades de los contornos no sólo detecta sino que también evita el problema de las oclusiones.

2.4.2.2. MÉTODOS PARA REDUCIR LA SENSIBILIDAD A LAS OCLUSIONES

El uso de métodos robustos es un camino para conseguir reducir la sensibilidad a las oclusiones en las correspondencias y otras diferencias en las imágenes. La presencia de oclusiones en pares de imágenes estéreo produce discontinuidades en la disparidad que por otro lado son coherentes. Es decir, existen regiones que por un lado tienen una discontinuidad grande en la disparidad, pero en otra dirección su función disparidad es suave. Esta suavidad introduce un nuevo umbral en la detección de oclusiones.

Otra aproximación para reducir la sensibilidad a las oclusiones es redimensionar la ventana de correlación para optimizar la similitud de las correspondencias cerca de las oclusiones.

2.4.2.3. MÉTODOS PARA MODELAR LA GEOMETRÍA DE LAS OCLUSIONES

Aunque los métodos anteriores para la detección y reducción de la sensibilidad a las oclusiones ofrecen distintas ventajas y todos son computacionalmente abordables, éstos no aprovechan todas las posibilidades que aportan las restricciones a la estéreo visión. Es deseable integrar el conocimiento de la geometría de las oclusiones dentro del proceso de búsqueda.

Belhumeur [26] define las bases de una serie de estimadores Bayesianos, cada uno de los cuales maneja un modelo más complicado del mundo. Los estimadores se

utilizan para definir funciones de coste para utilizar en programación dinámica. El modelo más simple supone que las superficies son suaves. El segundo asume que además de la suavidad de las superficies existen contornos abruptos en los objetos. El modelo tercero es más realista e incluye superficies con inclinaciones además de los contornos en los objetos. Variaciones de estos modelos, sobre todo del segundo, han sido utilizadas tanto en la programación dinámica como en los “graph cuts” para determinar el mapa de disparidad óptimo.

Otro método para detectar oclusiones y recuperar las profundidades de estas regiones es explotar la posibilidad de tener varias cámaras.

También se puede utilizar visión activa para detectar oclusiones y recuperar la profundidad de éstas, en base al estudio del movimiento de las cámaras, para conseguir que el punto oculto pase a ser visible.

2.4.3. APLICACIONES Y OTRAS CONSIDERACIONES

Una de las principales aplicaciones, hoy en día, para lo cual es necesario la estimación de un mapa denso de disparidades para una escena es la descomposición de una imagen en capas de igual profundidad para su posterior procesamiento y generación de nuevas vistas (*Image Based Rendering*) y en la reconstrucción tridimensional de un objeto a partir de varias vistas o una secuencia de video.

Otras aplicaciones pueden ser utilizar las imágenes del mapa de disparidad para estimar movimientos rígidos de los objetos en la escena tridimensional.

Otra aplicación de un mapa de disparidad denso se encuentra en la segmentación de video. Utilizando la información de profundidad se logra segmentar la escena en objetos que se encuentran a distintas profundidades en la misma. Una de las restricciones mayores es que la resolución de la disparidad puede no ser suficiente como para lograr separar objetos diferentes, si se encuentran a una profundidad relativamente parecida. Esto lleva a que esta aplicación se restrinja a escenas con una configuración determinada, con los objetos de interés relativamente separados en planos paralelos a la cámara para poder distinguirlos sin mucho error. La mezcla de la disparidad con otras características (color, posición, movimiento, textura, etc.) podría robustecer esta segmentación.

Los mapas de disparidad también son utilizados en vigilancia y conteo de personas en espacio cerrados y abiertos; en creación de realidad virtual, etc.

2.5. POSIBLES PROBLEMAS ASOCIADOS

Como hemos visto en el apartado referente a homografías, el proceso homográfico, en el caso ideal, puede convertir la perspectiva de una imagen tomada

de una escena en otra perspectiva equivalente a la imagen tomada por otra cámara. Pero comprendiendo mejor este proceso y observando la Figura 5 podemos ver que esta transformación sólo es ideal para el caso en el que la escena en cuestión sea una superficie plana. Por ello el principal problema de este proyecto es ver el resultado de esta operación en escenas con profundidad e intentar resolverlos. También debemos asumir que el resultado de antemano no podrá ser óptimo pues está el problema de los puntos ocluidos. Debido a que la escena tiene profundidad tendremos puntos que se vean en una imagen y que no se vean en la otra y viceversa. La cuestión es qué hacer con estos puntos en el momento de la transformación.

Otros problemas vienen asociados a la hora de escoger las parejas de puntos homólogos. Este proceso es crítico y aunque los algoritmos expuestos dan buenos resultados no siempre son perfectos. Debemos ver los resultados que obtenemos y analizar los escenarios donde se pueden aplicar y bajo que condiciones.

El tercer gran problema viene dado por los mapas de disparidad. Al igual que la parte de los puntos homólogos los resultados de los algoritmos que hemos visto no son ideales. Se acercan mucho a los mapas de profundidad reales pero no son perfectos. La principal consecuencia de esto es que al separar las imágenes en capas habrá puntos que se asignen a una capa a la que no pertenecen. Es decir tendremos capas que tendrán puntos de más y capas con puntos de menos. La inconveniente de este hecho es que al realizar las homografías por capas, se obtendrán resultados incorrectos para los puntos que no pertenezcan a una capa determinada, o para todos si esos puntos son precisamente los escogidos para calcular la homografía.

Todos estos problemas son los que se van a intentar solventar. Asimismo, en el desarrollo del proyecto han ido surgiendo otros problemas de ámbito menor que se han ido tratando sobre la marcha y se comentarán en el apartado que corresponda.

3. DETECCIÓN AUTOMÁTICA DE PUNTOS HOMÓLOGOS

3.1. BASADA EN EL DETECTOR DE ESQUINAS DE SHI Y TOMASI.

En este capítulo se explicará con detalle uno de los dos métodos implementados durante la realización del proyecto para conseguir correspondencias entre imágenes: el detector de esquinas propuesto por J. Shi y C. Tomasi [15]. La Figura 17 muestra los pasos que se han seguido para ello.

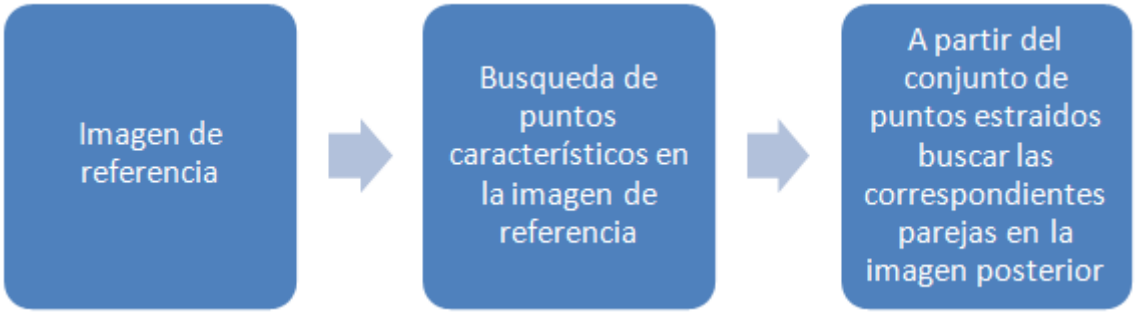

FIGURA 17 : PASOS A SEGUIR EN EL ALGORITMO DE SHI Y TOMASI

3.1.1. DEFINICIÓN

Un gran número de algoritmos utilizan como referencia para el *matching* la detección de bordes o *edges*. Aunque éstos no son sensibles a cambios de intensidad, presentan problemas cuando se presentan otras transformaciones entre imágenes. Un ejemplo de ellos es el detector de *Canny* [9].

Al encontrar bordes cercanos al umbral de detección, un pequeño cambio en la intensidad del mismo puede causar un gran cambio en su topología. Por tanto, sería muy probable un error en la búsqueda de correspondencias.

FIGURA 18 : IZQUIERDA: IMAGEN ORIGINAL. DERECHA: DETECCIÓN DE BORDES DE CANNY.

Para evitar ese efecto, el detector de *Shi y Tomasi* se basa, a diferencia de *Canny*, en la búsqueda de esquinas. Éstas son puntos característicos muy poco susceptibles a cambios de rotación y escala. Una esquina o *corner* se caracteriza por ser una región de la imagen con cambios de intensidad en diferentes direcciones. Éste será el principio básico de búsqueda de puntos de *Shi y Tomasi*. Filtrando la imagen con una ventana móvil en ocho direcciones, se obtienen tres tipos de región (Figura 19).

- *Flat* o plana: No hay cambios en ninguna dirección.
- *Edge* o borde: No hay cambios en la dirección del propio *edge*.
- *Corner* o esquina: Hay cambios significativos en todas direcciones.

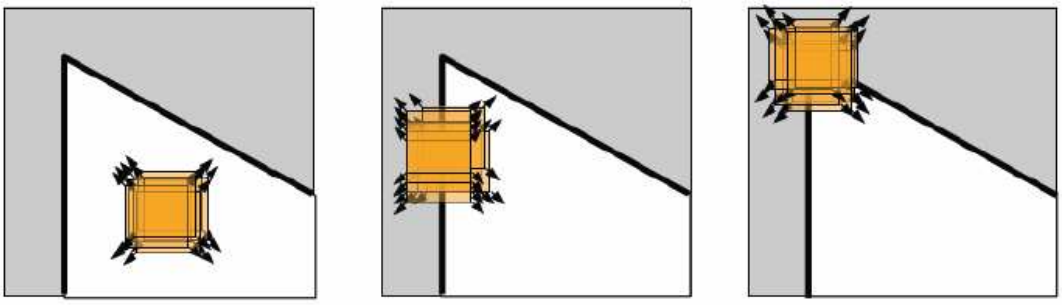

FIGURA 19 : TIPOS DE REGIONES DETECTADAS. DE IZQ. A DERECHA: FLAT, EDGE Y CORNER.

Una vez detectados los puntos de interés, en este caso los *corners*, se deben buscar las correspondencias entre puntos. En apartados posteriores se precisarán cómo se han calculado estas correspondencias.

3.1.2. BÚSQUEDA DE ESQUINAS

3.1.2.1. APLICAR EL GRADIENTE

El primer paso del algoritmo es calcular la matriz de autocorrelación 2x2 de la imagen a procesar. Para ello, previamente se obtienen las derivadas horizontal y vertical de primer orden para cada punto de la imagen, es decir el gradiente de cada punto. La idea es analizar los autovalores de la matriz de autocorrelación y así saber si un punto en concreto es una esquina, un borde o nada.

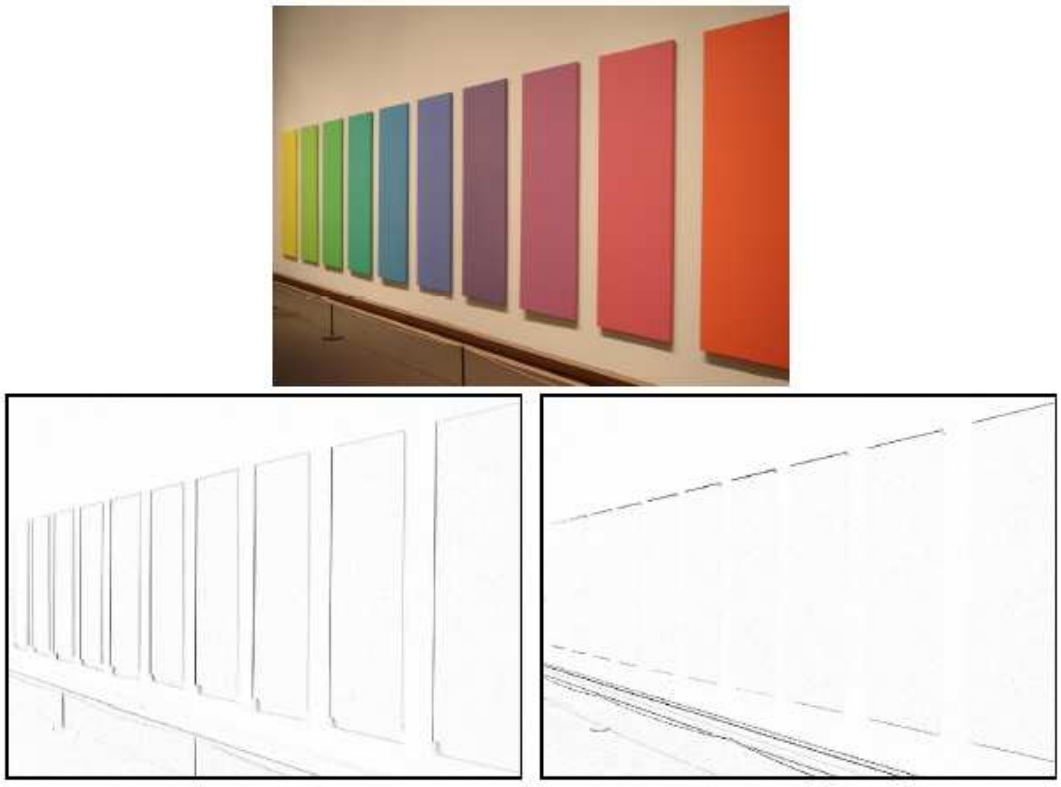

FIGURA 20 : ARRIBA: IMAGEN ORIGINAL. ABAJO IZQUIERDA: DERIVADA HORIZONTAL.

Se calculan los elementos de la matriz de autocorrelación.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$ Ec. 21

Donde

$$A = \left(\frac{\partial I}{\partial x}\right)^2 = X^2$$ Ec. 22

$$B = \left(\frac{\partial I}{\partial y}\right)^2 = Y^2$$ Ec. 23

$$C = (X * Y)$$ Ec. 24

Los elementos de la matriz de autocorrelación se obtienen al elevar al cuadrado las derivadas parciales.

Si definimos λ_1 y λ_2 como los valores propios (autovalores) de la matriz M calculada, se podrán obtener las tres tipos de regiones comentadas en el apartado 3.1.1.

- Si ambos valores son pequeños, indica que la función de autocorrelación es plana, por tanto la zona de la imagen tiene una intensidad aproximadamente constante → *Flat*
- Si uno de los valores es pequeño y otro es elevado, la función de autocorrelación tendrá un cierto rizado → *Edge*
- Si los dos valores son elevados, en la función se observarán picos bruscos → *Corner*

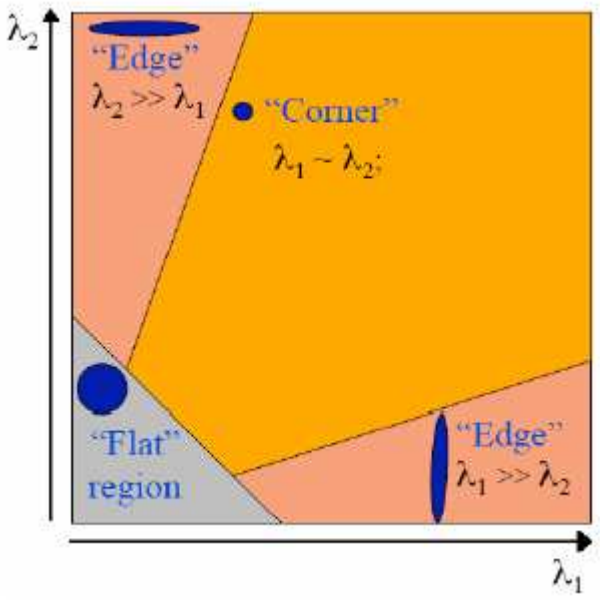

FIGURA 21 : REGIONES EN FUNCIÓN DE VALORES PROPIOS DE M.

Se puede justificar intuitivamente el motivo de esta clasificación a partir de los valores propios. Los valores de λ_1 y λ_2 reflejan los modos de variación de las direcciones principales de los gradientes. Por ese motivo, cuando en una región de la imagen los dos valores son elevados, deducimos que localmente existen dos direcciones importantes de los gradientes, y se concluye en que es una esquina.

En realidad el algoritmo de Shi y Tomasi a diferencia del de Harris sólo comprueba que uno de estos dos autovalores (el de menor valor) es mayor que cierto umbral si el autovalor mínimo es mayor a este umbral se asume que el punto en cuestión es una esquina.

3.1.2.2. NON-MAXIMAL SUPRESSION

Ésta es la última fase en la búsqueda sobre la imagen. En este caso ya no se buscan más puntos, sino que se trata de descartar varios de los obtenidos anteriormente.

Como ya mencionamos el primer paso es definir un umbral para descartar los píxeles que aparecen con el valor, de alguno de sus autovalores asociados, pequeño. Cuanto mayor sea este valor, más restrictivo será el detector en cuanto a número de *corners* detectados, aunque aumentará su fiabilidad.

Para evitar múltiples detecciones en una misma esquina (nubes de puntos negros) se utiliza el denominado filtro *non-maximal supression*. Este filtro se encarga de eliminar todos los puntos en los cuales la dirección del gradiente no sea la máxima en un entorno local.

Así tras la aplicación de este filtro, la supresión de las nubes de puntos, sólo quedarán píxeles individuales, que era el objetivo previo marcado**.**

3.1.3. CÁLCULO DE CORRESPONDENCIAS (MATCHING)

En este apartado se detalla el paso posterior a la búsqueda de puntos característicos que nos proporciona el detector, es decir, la descripción de las zonas de interés (vecindarios de puntos alrededor de los puntos críticos). De esta manera, podremos comparar descriptores entre pares de imágenes y buscar correspondencias entre ellas. Hay dos caminos posibles para ello. El primero seria hallar los puntos característicos de la imagen primera y su descriptor y hacer lo mismo con la segunda imagen. Así podemos comparar los descriptores de ambas imágenes y establecer las correspondencias entre puntos con algún tipo de medida. La otra forma es obtener los puntos característicos de la primera imagen junto con su descriptor. Y posteriormente comparar este descriptor con los puntos de la segunda imagen donde creemos que estarán su par correspondiente. La ventaja de la segunda opción es que te aseguras, en el caso de encontrar una correspondencia con alto valor, de que realmente sea su

punto homólogo. Con este criterio elegimos la segunda opción como método para el cálculo de correspondencias

Hay que señalar que dichos métodos no pertenecen estrictamente al detector de *Shi* y *Tomasi*, ya que la fase de 'detección' es independiente de la de 'descripción'.

Para entender el método escogido podemos observar la Figura 22.

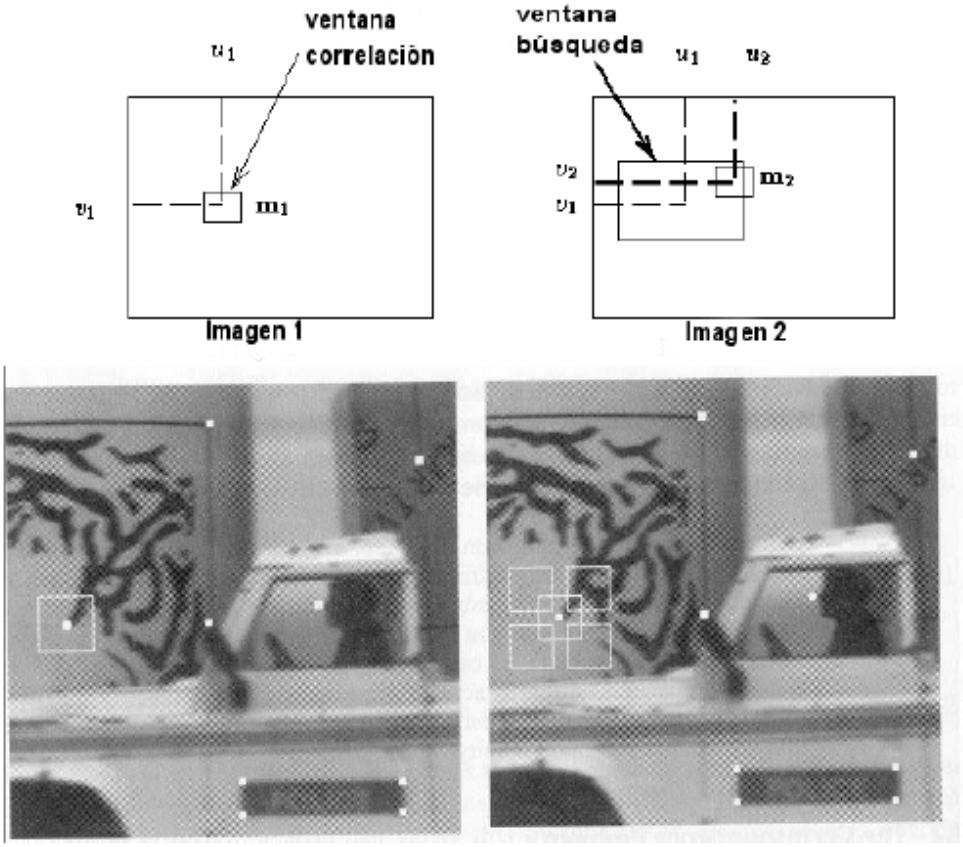

FIGURA 22 : EJEMPLO DE CORRESPONDENCIAS

Una vez que tenemos un punto característico de la imagen primera, hallamos su descriptor. En este caso una ventana centrada en el punto en cuestión. Posteriormente hemos de buscar en la imagen segunda algún punto cuyo descriptor se asimile de alguna manera al de la primera imagen.

Para la medida de similitud entre el descriptor de un punto de la primera imagen y los descriptores de los posibles puntos correspondientes de la segunda imagen, se pueden usar varias funciones. Dos funciones ampliamente usadas son:

a) ψ **(u,v)=uv** que da la correlación cruzada entre la ventana de la izquierda y la región de búsqueda en la imagen derecha.

$$r = \frac{\sum g_r g_b - n\,\overline{g_r g_b}}{\sqrt{\left(\sum g_r^2 - n\,\overline{g_r^2}\right)\left(\sum g_b^2 - n\,\overline{g_b^2}\right)}} \qquad \text{Ec. 25}$$

b) ψ $\mathbf{(u,v)=-(u-v)^2}$, que calcula la suma de los cuadrados de las diferencias o emparejamiento por bloques. Desarrollando esta última se puede poner de manifiesto la relación que existe entre ambas funciones.

Entre las dos opciones hemos optado por la primera pensando que aunque el coste computacional pueda ser elevado puede darnos resultados mas acertados.

La idea es establecer un umbral para dicha función. Establecido el umbral se empieza la búsqueda en la imagen segunda en la zona que creamos que se puede encontrar el píxel buscado. Una vez que la función de similitud nos de un resultado mayor o igual al umbral escogido asumiremos que el par de píxeles con los que estamos trabajando son un par de puntos homólogos.

3.2. IMPLEMENTACIÓN BASADA EN EL DETECTOR DE SHI Y TOMASI.

En esta fase del proyecto se tratará de establecer correspondencias entre puntos homólogos de dos imágenes. Para ello primeramente debemos detectar puntos característicos en la primera imagen y a continuación hallarlos en la segunda imagen.

Para realizar nuestro propio algoritmo hemos establecido inicialmente un número fijo de pares de puntos homólogos que debemos hallar para poder establecer posteriormente las homografías. Este número en concreto es nueve. Así hemos dividido la imagen en nueve cuadrantes y hemos hallado un único par de puntos homólogos por cuadrante. La idea de dividir la imagen en nueve cuadrantes se debe a poder obtener pares de puntos homólogos que no tengan las mismas coordenadas *x* e *y*. Además conseguimos establecer el emparejamiento con pares de puntos que corresponden con la globalidad de la escena. Es decir, si no dividiésemos la imagen en cuadrantes tras la ejecución del algoritmo de búsqueda de puntos característicos podríamos tener los puntos muy juntos debido a que en la imagen de búsqueda hubiesen esquinas muy juntas. La homografía (H) que obtuviésemos de estos emparejamientos podría no corresponderse con la globalidad de la imagen. Por todo ello decidimos dividir la imagen en cuadrantes. La razón de elegir el número de nueve cuadrantes viene del hecho físico de dividir la imagen en cuadrantes. Partiendo de la imposición de escoger como mínimo cuatro pares de puntos homólogos (como vimos en la sección correspondiente a las homografías) las posibles divisiones de una imagen empezarían con un tamaño de 1x4 o 2x2, es decir cuatro cuadrantes como mínimo. Como en un principio no sabemos como va a ser la captura con la que vamos a trabajar nos parece buen criterio dividir la imagen de tal manera que el número de cuadrantes sea el mismo horizontal y verticalmente. Así las posibilidades que tenemos son de 2x2, 3x3, 4x4, 5x5…La configuración de 4x4 establece 16 divisiones. Esto se puede convertir en un problema si la imagen no esta muy texturizada. Podrían existir varios puntos elegidos como característicos pero que no lo fuesen. La opción de 2x2 tampoco nos satisface debido al número de emparejamientos. Son escasos. Por ello hemos decidido la configuración de 3x3, es decir, nueve divisiones. Esto nos daría nueve pares de puntos homólogos repartidos por toda la imagen.

También hemos tenido en cuenta que en las imágenes podrían existir puntos móviles. Por ello hemos desarrollado un método inicial de descarte de estos puntos móviles.

Para entender mejor el desarrollo del algoritmo realizado presentamos el siguiente diagrama de bloques.

FIGURA 23 : DIAGRAMA DE BLOQUES DEL DESARROLLO DE NUESTRO ALGORITMO

Tras la presentación de este diagrama de bloques que describe nuestro algoritmo procedemos a explicar detenidamente cada parte.

3.2.1. PRIMER PASO: FILTRADO DE PUNTOS EN MOVIMIENTO

En la implementación de nuestro algoritmo lo primero que hemos tenido en cuenta es extraer de las imágenes los posibles puntos móviles. Así en la posterior búsqueda de los puntos homólogos minimizaríamos los posibles errores de emparejamientos.

Se trata de establecer un umbral por encima del cual los puntos de la imagen diferencia resultante entre dos imágenes de una misma cámara puedan ser evaluados como parte de un conjunto de puntos móviles.

La idea es hallar la imagen resultante de la resta de dos imágenes consecutivas de una misma cámara y establecer un umbral (T). Así los puntos de la

imagen resta que estén por encima de este umbral se consideraran puntos móviles. Esto se hace para las dos cámaras, es decir, necesitamos un mínimo de cuatro imágenes. Dos por cada cámara.

Para hallar estos puntos se utiliza un algoritmo automático de umbralización. Este algoritmo haya el valor T. Todos los puntos de *imagen resta* que tengan un valor superior a T se consideraran candidatos a puntos móviles.

El valor T se haya sobre toda la *imagen resta* y posteriormente se va evaluando la *imagen resta* en nueve bloques individuales para detectar la posible aparición de varios puntos móviles.

Sobre estos puntos se aplica una técnica de erosión para eliminar puntos aislados con una máscara:

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

Una vez eliminados se aplica una técnica de dilatación para ampliar las zonas en movimiento. Luego se extraen las coordenadas máxima y mínima de *x* e *y* en movimiento de cada bloque.

El resultado sería, por cada bloque, cuatro coordenadas *x1, x2, y1, y2* entre las cuales se considera que hay movimiento. Si *x1=x2=y1=y2 =1* se considera que en dicho bloque no hay movimiento.

3.2.2. SEGUNDO PASO: RECORTE DE LA IMAGEN DE REFERENCIA

En este segundo paso vamos a ver como recortar la imagen de referencia para poder escoger buenos puntos para buscar correspondencias (*matching*).

El problema que intentamos resolver en esta etapa es el siguiente. En la imagen de referencia (imagen sobre la que se ejecutará el algoritmo de Shi y Tomasi), tras ejecutar el algoritmo de búsqueda de puntos característicos, podemos encontrarnos que un punto de los escogidos esté cerca del límite de la imagen. Si es

así puede que no encontremos su homólogo en la imagen posterior (imagen sobre la que se buscaran las correspondencias) porque sencillamente no este en la imagen. Con lo cual primero tenemos que ver cuales son los puntos límite de la imagen posterior y posteriormente recortar la imagen de referencia para buscar únicamente puntos que existan en las dos imágenes.

El algoritmo de búsqueda de puntos característicos utiliza la operación *gradiente* sobre cada punto de búsqueda. Para ello la operación *gradiente* necesita un entorno (bloque) centrado en el punto evaluado. Este tamaño de bloque es fijo para todos los puntos y se configura como parámetro al inicio del algoritmo. También se introduce como parámetro el tamaño del desplazamiento que se realizará en la posterior búsqueda de correspondencias. En la imagen posterior y tras haber extraído los puntos característicos de la imagen de referencia debemos buscar los puntos correspondientes. Por ejemplo, imaginemos el punto característico x_i encontrado en la imagen de referencia, con coordenadas *coordenada_x* y *coordenada_y*. La búsqueda en la imagen posterior de su punto correspondiente se realizará a partir de las coordenadas *coordenada_x* y *coordenada_y*, en un entorno de *desplazamiento_x* - *coordenada_x* hasta *desplazamiento_x* + *coordenada_x* y *desplazamiento_y* – *coordenada_y* hasta *desplazamiento_y* + *coordenada_y*.

Tenemos la altura y la anchura del bloque sobre el que aplicaremos el gradiente *y_bloq* y *x_bloq*. Además tenemos el desplazamiento lateral y vertical que se realizará en la imagen posterior para hallar el punto correspondiente de la imagen posterior mediante correlación.

Con la Figura 24 podremos entender mejor este concepto.

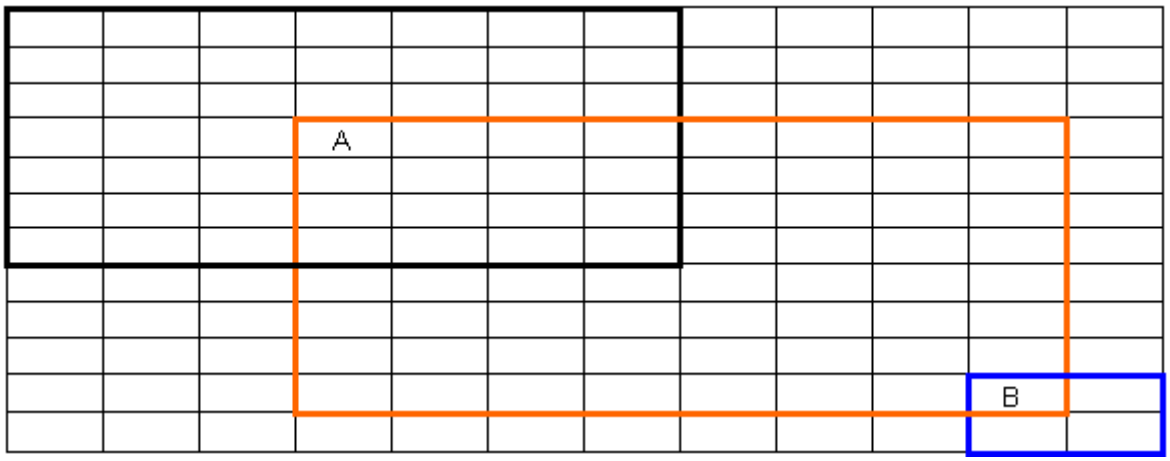

FIGURA 24 : ESQUEMA DEL RECORTE QUE SUFRIRÁ LA IMAGEN.

Supongamos que este cuadro muestra la esquina superior izquierda de la imagen posterior sobre la que vamos a tratar de hallar un punto dado mediante correlación. Recordamos que el tamaño de bloque que se usa para aplicar el gradiente en la imagen de referencia es el mismo tamaño de bloque que se usa para aplicar la correlación en la imagen posterior. El punto A correspondería al punto límite sobre el que se puede extraer un bloque de *y_bloque* x *x_bloque* (7 * 7) de la imagen. Este punto correspondería al desplazamiento máximo tanto en *y* como en *x* desde el punto

límite de la imagen de referencia sobre el que hemos buscado un punto característico. Con lo cual retrocediendo *desplazamiento_y* =7 y *desplazamiento_x* =7 llegamos al punto B que es el punto límite de la imagen de referencia sobre el cual podemos buscar un punto característico.

Es decir, el recorte de la imagen de referencia es igual a:

Limite_y = desplazamiento_y + ceil (*y_bloque* /2) - 1

Limite_x = desplazamiento_x + ceil (*x_bloque* /2) -1

3.2.3. TERCER PASO: SELECCIÓN DE LOS PUNTOS CARACTERÍSTICOS

Una vez recortada la imagen de referencia la pregunta es ¿cómo escoger esos puntos característicos?

La selección de dichos puntos la hemos implementado a partir del algoritmo propuesto por Shi y Tomasi [15]. La descripción de este método ya la hemos desarrollado anteriormente. Como comentábamos en la sección anterior el tamaño del bloque que utiliza la operación del gradiente es un parámetro que se le debe configurar al inicio del algoritmo. El tamaño establecido de forma general y después de observar los resultados es de 9x9, es un resultado empírico que puede variar según la imagen tratada. La única diferencia que hemos adoptado respecto del algoritmo de Shi y Tomasi está a la hora de escoger los puntos entre los posibles candidatos.

Según el algoritmo propuesto la selección de los puntos viene determinada por un umbral. Si el autovalor asociado de menor valor supera dicho umbral el punto en cuestión seria un posible candidato. Así tras la ejecución del algoritmo tendríamos una serie de posibles candidatos dependientes del umbral. A medida que el umbral aumentase en valor, el número de puntos seleccionados debería de decrecer. Nuestra alternativa ha sido la siguiente: hemos partido de haber dividido la imagen en nueve cuadrantes para escoger únicamente nueve pares de puntos, uno por cuadrante. Así pues nuestra solución ha sido aplicar el algoritmo de Shi y Tomasi a cada cuadrante y no tener en cuenta el umbral, es decir, simplemente escogemos el punto cuyo valor del autovalor mínimo asociado sea el máximo. Esa es la única diferencia respecto al algoritmo original. Con ello tras la ejecución del procedimiento obtenemos nueve puntos característicos, los más característicos según su cuadrante.

Somos conscientes de que esta implementación no es óptima. Se podría seleccionar un punto característico en un cuadrante que no fuera precisamente “característico”. Esto podría ocurrir por ejemplo si ese cuadrante en concreto solo tuviera una superficie sin texturizar (el cielo despejado por ejemplo). Cualquier punto de ese cuadrante no sería válido pero como no se tiene en cuenta ningún umbral, se escogería un punto como característico. Por ello para el buen funcionamiento del algoritmo las imágenes deben estar bien texturizadas en los nueve cuadrantes. Así nos aseguramos de que los puntos seleccionados como característicos lo son verdaderamente.

3.2.4. CUARTO PASO: LOCALIZACIÓN DE PUNTOS HOMÓLOGOS

Como anticipamos anteriormente, el método escogido para detectar los puntos correspondientes de la imagen posterior ha sido la correlación de una ventana centrada en el píxel de la imagen de referencia con las ventanas de los posibles píxeles que pueden ser los correspondientes en la imagen posterior.

La función de correlación usada es la siguiente:

$$r = \frac{\sum g_r g_b - n\,\overline{g_r g_b}}{\sqrt{\left(\sum g_r^2 - n\,\overline{g_r^2}\right)\left(\sum g_b^2 - n\,\overline{g_b^2}\right)}} \qquad \text{Ec. 26}$$

Donde

n: número de píxeles de las matrices de referencia y de búsqueda.

g_r: cada uno de los valores de la matriz de referencia.

g_b: cada uno de los valores de la matriz de búsqueda.

$\bar{g}_r$: valor medio de los valores de la matriz de referencia.

$\bar{g}_b$: valor medio de los valores de la matriz de búsqueda.

r: valor de correlación. Su valor absoluto está comprendido entre 1 (cuando las dos matrices son idénticas) y 0 (cuando no hay correlación entre ellas).

El método desarrollado es sencillo: se calcula el valor de correlación, r, con la ventana de correlación centrada en el píxel de referencia y las distintas ventanas de correlación centradas en los posibles candidatos de la imagen posterior. Para elegir a estos candidatos lo que hemos establecido es una ventana de búsqueda centrada en la posición que indica el píxel de la imagen de referencia pero en la imagen posterior.

Llegados a éste punto tenemos que aclarar las diferencias entre la ventana de correlación y la ventana de búsqueda. La ventana de búsqueda está centrada sobre las coordenadas del píxel de referencia pero en la imagen posterior. Contiene los puntos que se evaluarán mediante la técnica de correlación en la imagen posterior para seleccionar el punto correspondiente con el punto característico de la imagen de referencia. La otra ventana, la de correlación es la ventana que usa la fórmula de la correlación para establecer los emparejamientos. Para cada punto de la imagen posterior que sea posible candidato, se extrae un bloque centrado en éste punto del tamaño de la ventana de correlación.

El tamaño de esta ventana de búsqueda queda determinado por los parámetros *desplazamiento_x* y *desplazamiento_y*. Como mencionamos en la sección 3.2.2 el tamaño es de (2*desplazamiento_x + 1) x (2*desplazamiento_y + 1). El

tamaño de la ventana de correlación es el mismo que el tamaño del bloque que necesita la operación del gradiente de la sección 3.2.3 en nuestro caso es de 9x9.

Así si el píxel de referencia esta en la posición i,j lo que se hace es obtener una ventana de búsqueda de la imagen posterior centrada en esta posición. Y para cada píxel de esa ventana se calcula el valor de *r* respecto al píxel de referencia (búsqueda exhaustiva). Después se busca el píxel que ha obtenido el mayor valor de correlación y si no supera un umbral definido próximo a 1 se amplia la ventana de búsqueda hasta encontrar algún emparejamiento que supere este umbral. Esta ampliación se realiza hasta un número límite de veces. Nuestro límite esta establecido en 100, se puede ampliar pero entonces se aumenta el coste computacional. Este parámetro no es tan trivial como parece, pues si se configura muy pequeño y el desplazamiento también es pequeño se corre el riesgo de no encontrar ningún emparejamiento correcto. Este parámetro se puede entender como una segunda oportunidad de encontrar un buen emparejamiento si tras buscar en los puntos candidatos no se ha encontrado ninguno bueno. Es decir, debemos establecer el los parámetros de desplazamiento de forma coherente con los posibles desplazamientos que nos podamos encontrar entre las imágenes.

Pongamos un ejemplo para entender este concepto. Imaginemos que la correspondencia del punto (*x1,y1*) de la imagen de referencia (imagen izquierda) es el punto (*x2,y2*) de la imagen posterior (imagen derecha). Supongamos que están sobre la misma línea epipolar, tendríamos *y1 = y2* y *x1+d =x2.* Si escogemos un tamaño de desplazamiento de 1, tendríamos una ventana de búsqueda de (2*1+1) x (2*1+1) es decir, de 3x3. Si además escogemos un número límite de ampliación de 1, la ventana de búsqueda (en la imagen posterior) se podría ampliar hasta un tamaño de 5x5 centrada en el punto (*x1,y1*). Sabiendo que *x1 +d =x2* podemos asegurar que para una d igual o mayor a 3 el emparejamiento será erróneo, pues la búsqueda se hará como mucho en una ventana de 5x5.

Debemos resaltar que aunque en la fase de detección de puntos característicos hemos dividido la imagen en nueve cuadrantes, en la fase de búsqueda de correspondencias no se tienen en cuenta los nueve cuadrantes. Es decir, se puede encontrar un punto correspondiente en la imagen posterior que no pertenezca al cuadrante en el cual está enmarcado el punto característico de la imagen de referencia.

El principal parámetro que debemos configurar en éste momento sería el desplazamiento de los puntos que nosotros estimemos necesario. Para valores pequeños hemos comprobado que puede dar errores de emparejamiento. Para altos valores el coste computacional incrementa considerablemente. Por ello si después de ampliar un número límite de veces la ventana de búsqueda no se encuentra ningún emparejamiento que supere el umbral se da por concluida la búsqueda y se concluye que el emparejamiento establecido puede ser erróneo. No es del todo fiable. Esto no asegura tampoco que el emparejamiento sea erróneo porque la fórmula de la correlación no es exacta, pero si nos puede hacer sospechar que quizás ese

emparejamiento no sea correcto. La solución sería intentar obtener un par de imágenes más texturizadas.

3.2.5. QUINTO PASO: REPRESENTACIÓN GRÁFICA

En éste paso se ilustran ejemplos de resultados obtenidos. La imagen de referencia estará dividida en nueve cuadrantes y tendrá enmarcados sus nueve puntos característicos. La imagen posterior tendrá enmarcadas las nueve correspondencias encontradas. Si alguna fuese “no fiable” estaría enmarcada con un marco aleatorio como se observa en la siguiente Figura 25.

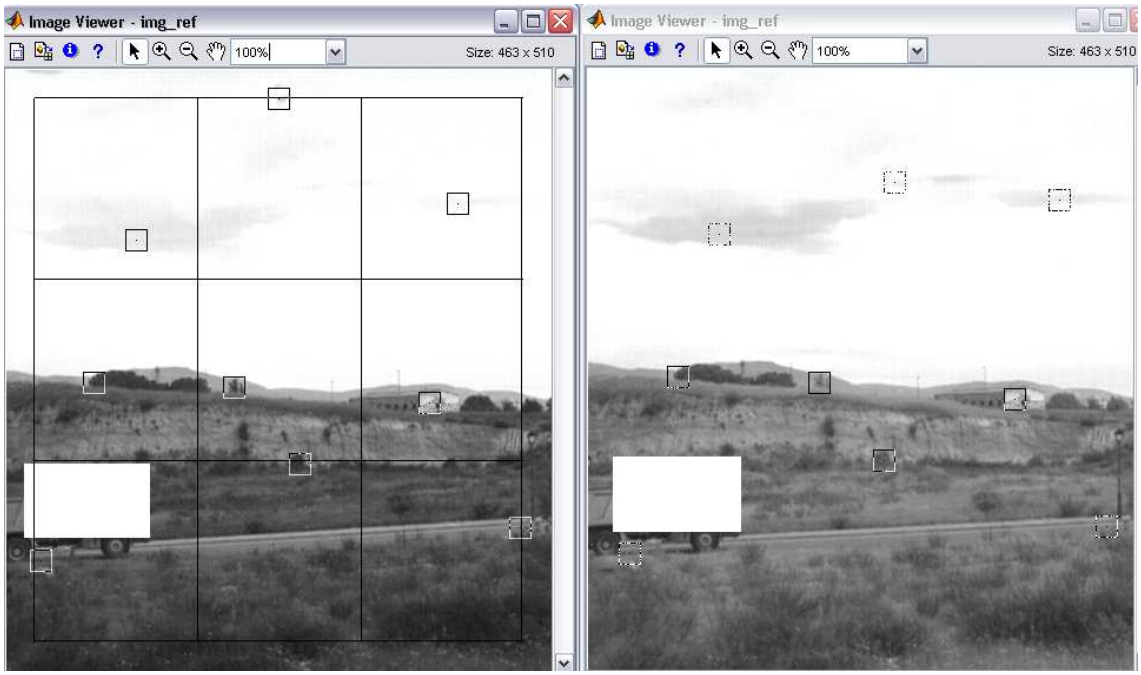

FIGURA 25 : LA IMAGEN DE LA IZQUIERDA CORRESPONDE CON LA IMAGEN DE REFERENCIA. LA DERECHA CORRESPONDE CON LA IMAGEN POSTERIOR.

Se puede observar como el único fallo corresponde con el cuadrante superior central. Además hay puntos como los de los cuadrantes superiores laterales que están enmarcados con un marco aleatorio (tiene puntos blancos y negros de forma aleatoria). Esto quiere decir que aunque a simple vista se observe que son buenos pares de puntos, su valor de correlación no ha superado el umbral de 0.9. Así podemos sospechar que a lo mejor no son buenos pares de puntos. También podemos observar el rectángulo en blanco que corresponde con posibles puntos en movimiento y donde no se escogerá ningún punto.

3.3. RESULTADOS DEL ALGORITMO DE SHI Y TOMASI

Todas las pruebas realizadas son sobre imágenes en escala de grises, así simplificamos los cálculos. Como hemos visto en la implementación de nuestro algoritmo, éste necesita tres parámetros para su ejecución. El primero es el tamaño del bloque para empezar a detectar los puntos característicos con el algoritmo de Shi y Tomasi. El segundo parámetro es el desplazamiento de las ventanas a la hora de buscar las correspondencias entre los puntos en la imagen posterior mediante el método de correlación. El tercero es el umbral de detección para el algoritmo de correlación, este parámetro esta configurado por defecto en 0.9 se puede dejar fijo. Si bajamos ese umbral podemos establecer un mayor número de emparejamientos erróneos. El segundo parámetro también podemos dejarlo fijo, pues si no encuentra ningún emparejamiento que supere el umbral se ampliara automáticamente este parámetro. Por ello el único parámetro a tener en cuenta es el primero, el tamaño del bloque que rodea al punto en cuestión. Sobre el que se aplicará el algoritmo de Shi y Tomasi.

La primera prueba que realizamos es para una imagen (la llamaremos "tsukuba") de 288x384 con varias profundidades no definidas claramente pero bastante bien texturizada. El tamaño del bloque es 11 x 11.

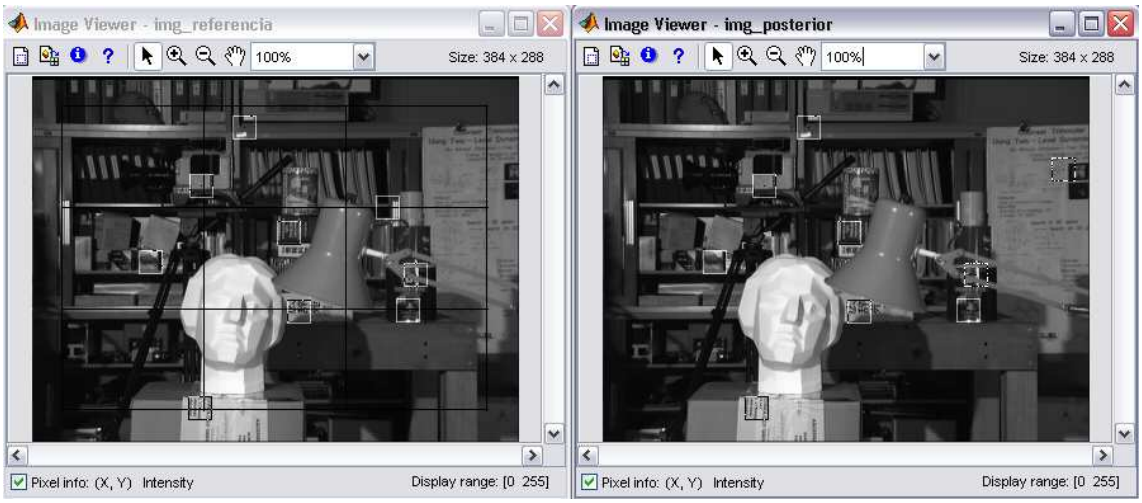

FIGURA 26 : "TSUKUBA" BLOQUE DE 11X11

Se puede observar que la única pareja de puntos homólogos que falla corresponde a la del cuadrante superior derecho. Esto es debido a que el cuadro de referencia (el descriptor del punto de referencia) tiene parte del fondo de la escena (la parte blanca del descriptor). En la imagen posterior esta parte del fondo está tapada por el bote y por eso a la hora de hallar la correspondencia se descarta ese punto: el error, por tanto, es debido a una oclusión.

Si bajamos el tamaño de bloque a un valor de 5x5 e invertimos la escala de grises (para ver mejor los puntos marcados) obtenemos el siguiente resultado.

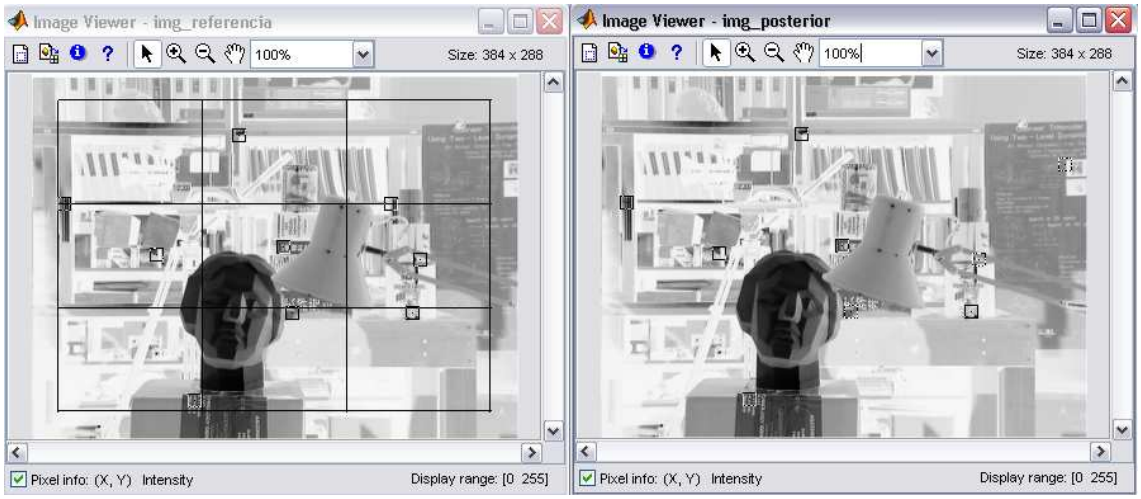

FIGURA 27 : "TSUKUBA" BLOQUE DE 5X5

El resultado es el mismo que con mayor tamaño de bloque.

Ahora reducimos el tamaño del bloque a 3x3.

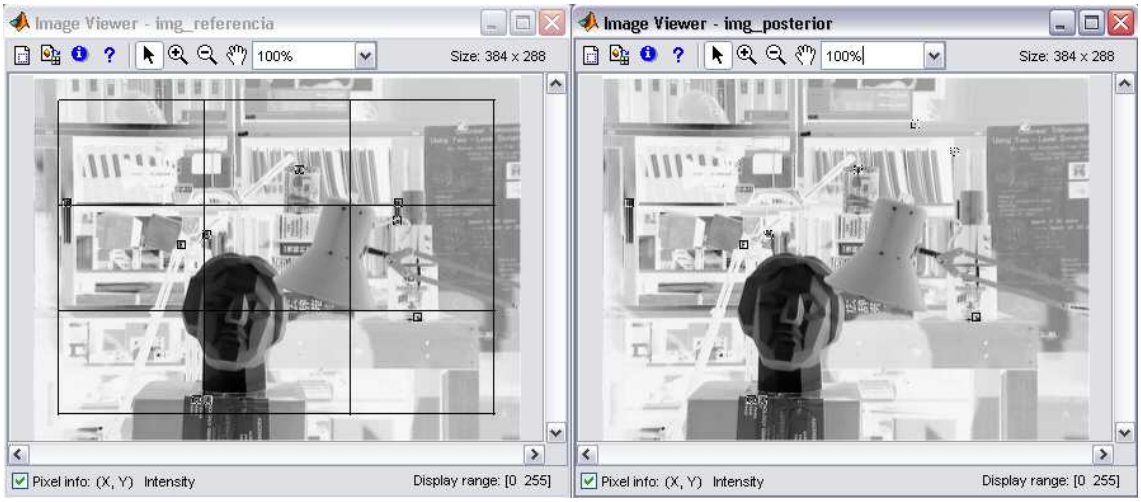

FIGURA 28 : "TSUKUBA" BLOQUE DE 3X3

Los cuadros que rodean los puntos seleccionados son muy pequeños pero podemos observar que los emparejamientos de los cuadrantes derechos, el medio y el superior son erróneos.

La segunda prueba la hacemos con una imagen (la llamaremos "cartas") de tamaño 207x384 con tres profundidades diferenciadas claramente, además está mejor texturizada que la anterior. El tamaño del bloque primeramente es de 11x11.

FIGURA 29 : "CARTAS" BLOQUE DE 11X11

Observamos que todos los emparejamientos son correctos. Se pueden ver dos de ellos en la carta del "as de oros" y otros tres en el "rey de picas".

Procedemos a reducir el tamaño de bloque directamente a 3x3.

FIGURA 30 : "CARTAS" BLOQUE DE 3X3

Podemos observar que los emparejamientos siguen siendo correctos, en este caso bajar el tamaño de bloque no ha afectado en el emparejamiento de los puntos.

Ahora probaremos con un par de imágenes (las llamaremos "camión") tomadas por dos cámaras distintas, no están rectificadas y poseen distinta iluminación. Además hay un objeto en movimiento. La profundidad no es muy significativa, es decir, no se observan a priori posibles puntos ocluidos. Y prácticamente la mitad superior de la imagen está mal texturizada. El tamaño de las imágenes es de 463x510. El tamaño del bloque es de 11x11 inicialmente.

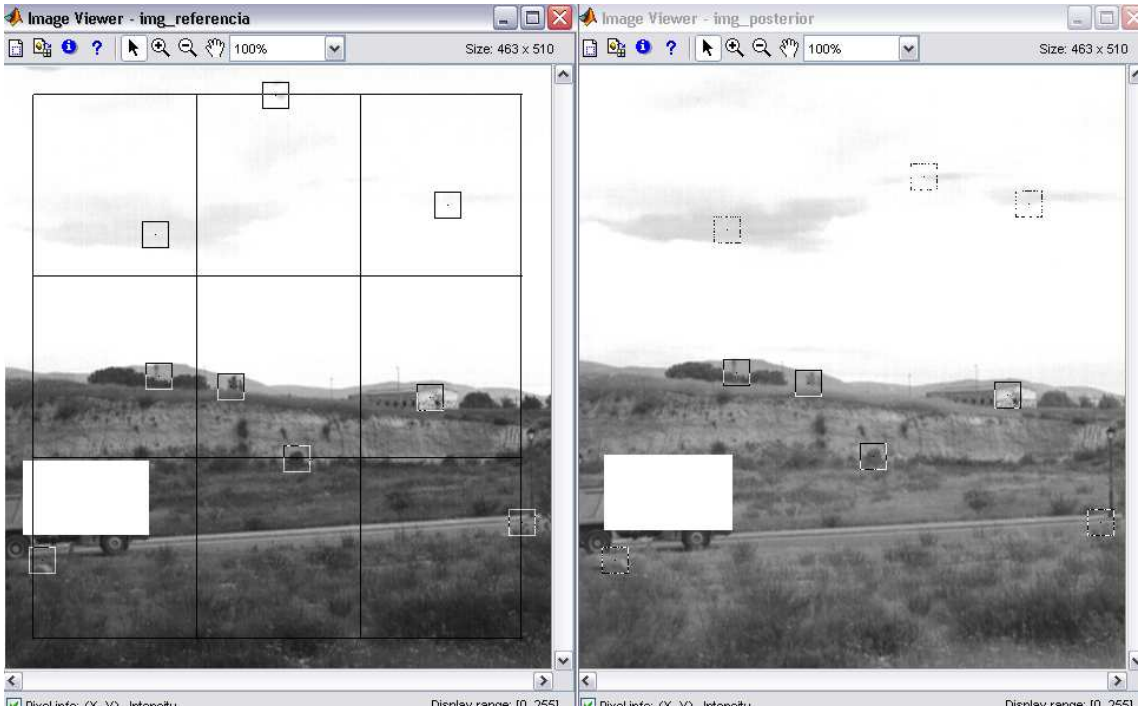

FIGURA 31 : “CAMIÓN” BLOQUE DE 11X11

Podemos ver que el único error es del que corresponde al cuadrante superior central. Además comprobamos como la parte móvil (el camión) es extraído de la imagen. Otro aspecto a tener en cuenta y que se ve mejor en éste par de imágenes que en las anteriores es el recuadro aleatorio que rodea a los emparejamientos de los cuadrantes: superiores e inferiores laterales. Como vimos en el desarrollo del algoritmo esto significa que el umbral de correlación no ha sido superado y pueden corresponderse con emparejamientos erróneos, aunque visualmente podamos ver que son correctos.

Ahora reducimos el tamaño del bloque a 3x3.

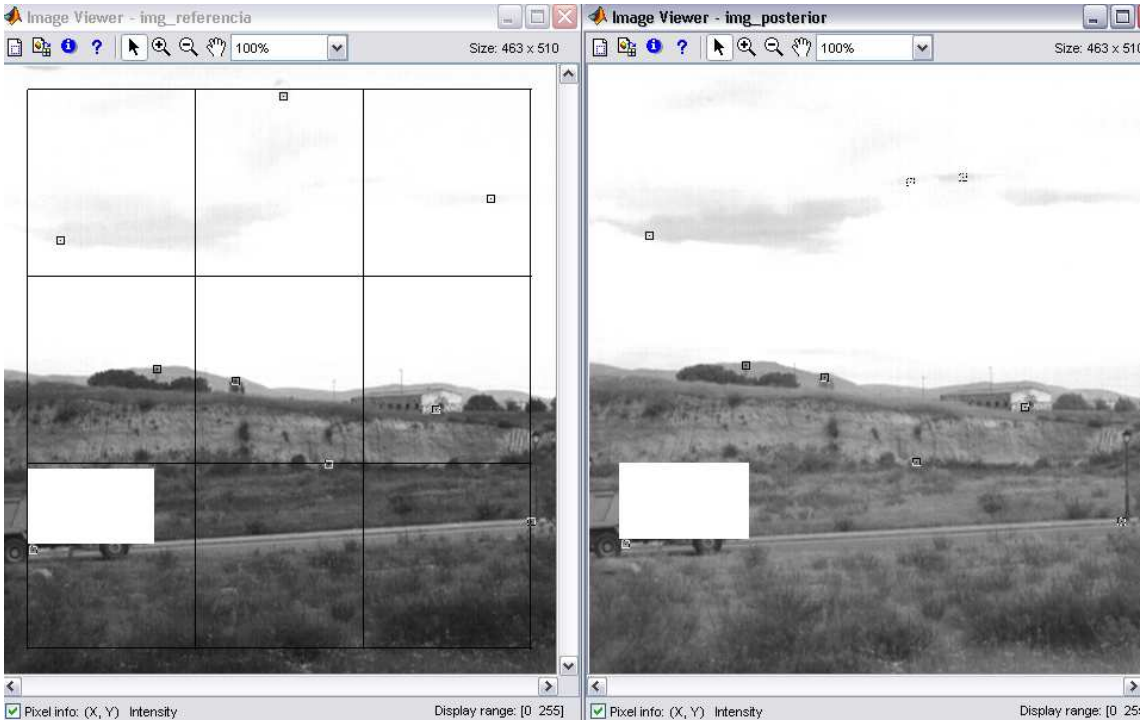

FIGURA 32 : "CAMIÓN" BLOQUE DE 3X3

Se observa que los emparejamientos fallidos corresponden a los cuadrantes superiores central y derecho. Que corresponden con los cuadrantes del cielo mal texturizado.

3.4. CONCLUSIONES DEL ALGORITMO DE SHI Y TOMASI

Aunque las pruebas realizadas no han sido exhaustivas, podemos concluir de forma general que el resultado del algoritmo es el esperado. Los principales objetivos están cumplidos: consigue establecer pares de puntos homogéneos y detecta las partes móviles de la escena.

Como hemos visto en la sección anterior referente a los resultados se deben cumplir una serie de requisitos. Empecemos viendo los fallos que se han encontrado para así extraer las condiciones de funcionamiento.

En general hemos observado que reduciendo el tamaño del bloque sobre el que se aplica el algoritmo de detección de puntos característicos aumentaba el número de puntos mal emparejados. En realidad solo ha aumentado en una pareja de puntos para las imágenes de "tsukuba" y "camión" pero es un factor a tener en cuenta.

Otro de los fallos se observa en las Figura 26, Figura 27 y Figura 28 (tsukuba). El problema del emparejamiento erróneo es que se selecciona un punto ocluido y que gran parte de su vecindad esta ocluida para la imagen posterior, por eso no se puede reconocer en la imagen posterior y lo empareja con otro punto. Es de esperar que

estos errores no se produzcan si la búsqueda de puntos homólogos se realiza sobre planos de igual profundidad.

Además en la imagen “camión” podemos ver que los fallos de emparejamiento se producen debido a que el cielo esta muy mal texturizado. No hay puntos característicos muy relevantes. Por eso en la búsqueda sobre la imagen posterior no se establecen buenos resultados.

Una vez vistos los principales problemas establecemos las siguientes condiciones para el correcto funcionamiento del algoritmo:

Profundidad: Si la escena tiene mucha profundidad aparecerán más puntos ocluidos. Si hay muchos puntos ocluidos se puede seleccionar uno de ellos o un punto cuyo entorno este ocluido y dar lugar a un mal emparejamiento.

Texturización: Si la escena está mal texturizada la selección de puntos característicos no será muy óptima. Además en la posterior búsqueda de correspondencias se pueden producir emparejamientos erróneos.

Tamaño del bloque de búsqueda: Si se reduce mucho se pueden elegir como puntos “buenos” puntos que en realidad no lo son. En la posterior búsqueda darían problemas.

Estas son las principales condiciones que se precisan para el óptimo funcionamiento del algoritmo.

3.5. TÉCNICA SURF

Tras desarrollar el proyecto hasta su última fase nos encontramos con un gran problema relacionado con éste punto. Según implementamos nuestro algoritmo de detección de puntos homólogos se escogían nueve pares de puntos homólogos repartidos por toda la imagen. Cuando llegamos a la fase final del proyecto, nos dimos cuenta de que al separar la imagen derecha en *n* capas incumplíamos el principal requisito de nuestro algoritmo de detección de puntos. Es decir, ocurría que más de un cuadrante no estaba texturizado debido a que había capas que contenían pocos puntos reales. Podemos ver en la siguiente figura una de las capas extraídas de la imagen derecha y su correspondiente imagen izquierda sobre la que se buscarán los puntos correspondientes.

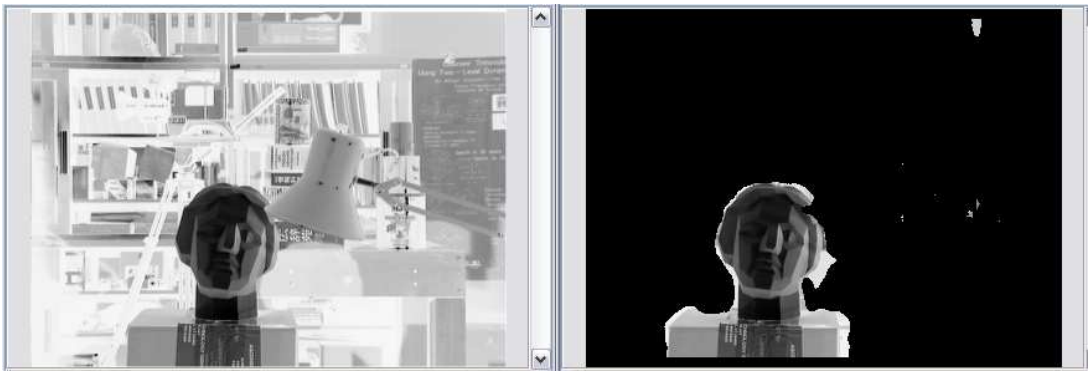

FIGURA 33 : IMAGEN IZQUIERDA Y CAPA DE LA IMAGEN DERECHA ENTRE LAS QUE SE PROCEDERÁ A ESTABLECER LOS PARES DE PUNTOS HOMÓLOGOS

Si dividimos la imagen de la derecha, es decir la capa del busto, en nueve cuadrantes podemos observar a simple vista que tendremos cuadrantes con todos los puntos negros. No se establecerán buenos emparejamientos. Por ello la idea de dividir la imagen en cuadrantes no es satisfactoria para nuestro propósito de establecer las homografías por capas.

Por éste motivo nos planteamos dos cosas: mejorar nuestro algoritmo para poder extraer puntos en cada capa sin tener en cuenta los cuadrantes o utilizar una técnica muy actual y robusta en la detección de puntos característicos y su posterior búsqueda de puntos correspondientes. Escogimos la segunda opción por dos razones: la mejora de nuestro algoritmo implicaba cierta incertidumbre a la hora de seleccionar puntos característicos. Es decir, una vez elegido un buen punto característico, para que el siguiente no esté muy cerca (no tenga igual la coordenada x ó y) debíamos elegir un radio mínimo donde no buscar puntos ¿qué radio elegir? Elegir un posible radio era algo bastante empírico, nuestro objetivo sería tener el mínimo de parámetros configurables. La segunda razón es la posible comparación entre los dos métodos y poder ver cual de ellos es mejor en las circunstancias dadas. Por ello elegimos la técnica de SURF.

Ésta técnica se denomina abreviadamente SURF (*Speeded Up Robust Features*). Vamos a proceder a hacer una breve descripción del método sin entrar demasiado en los detalles de su funcionamiento y posteriormente haremos unas pruebas para analizar los resultados y evaluar las condiciones de su uso.

3.5.1. DESCRIPCIÓN DEL ALGORITMO SURF

El algoritmo de SURF está basado en su predecesor SIFT. Éstas técnicas se basan en el concepto de *multi-resolución.* La técnica consiste en replicar la imagen original para así buscar puntos que estén en todas las réplicas. Con esto se asegura la invariancia a la escala. La técnica de *multi-resolución* se puede desarrollar de dos maneras, estableciendo las réplicas de manera piramidal (pirámide de Gauss o de Laplace) u obteniendo imágenes de igual tamaño pero reduciendo su ancho de banda. Para entender la forma piramidal observemos la siguiente figura.

FIGURA 34 : PIRÁMIDE DE GAUSS

La idea es producir una pirámide donde la siguiente imagen tenga la mitad del tamaño que la anterior. Para ello se debe filtrar primeramente la imagen inicial con un filtro paso bajo y después submuestrearla para no perder información en el proceso. Observar la Figura 35.

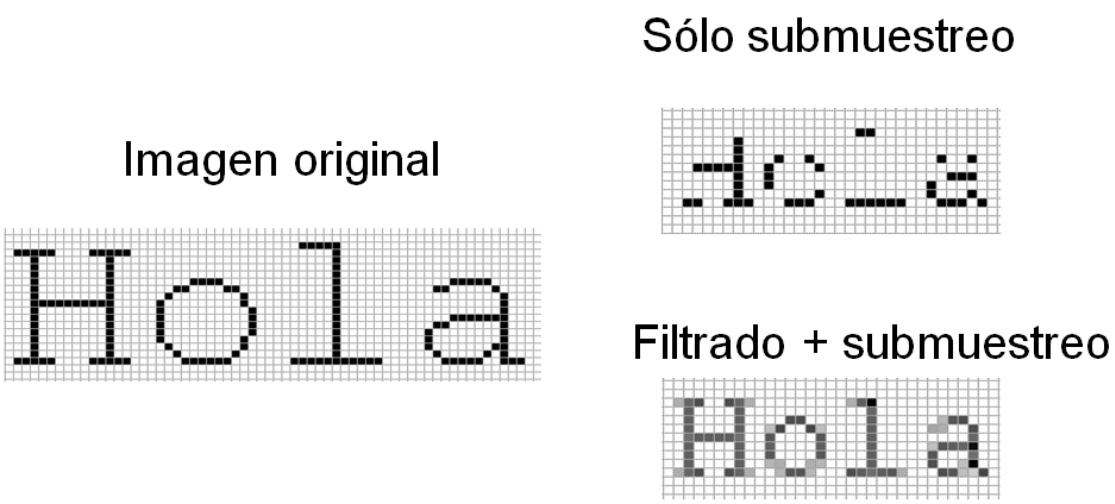

FIGURA 35 : PROBLEMA DEL SUBMUESTREO

La pirámide de Laplace es parecida a la de Gauss. Observar la Figura 36.

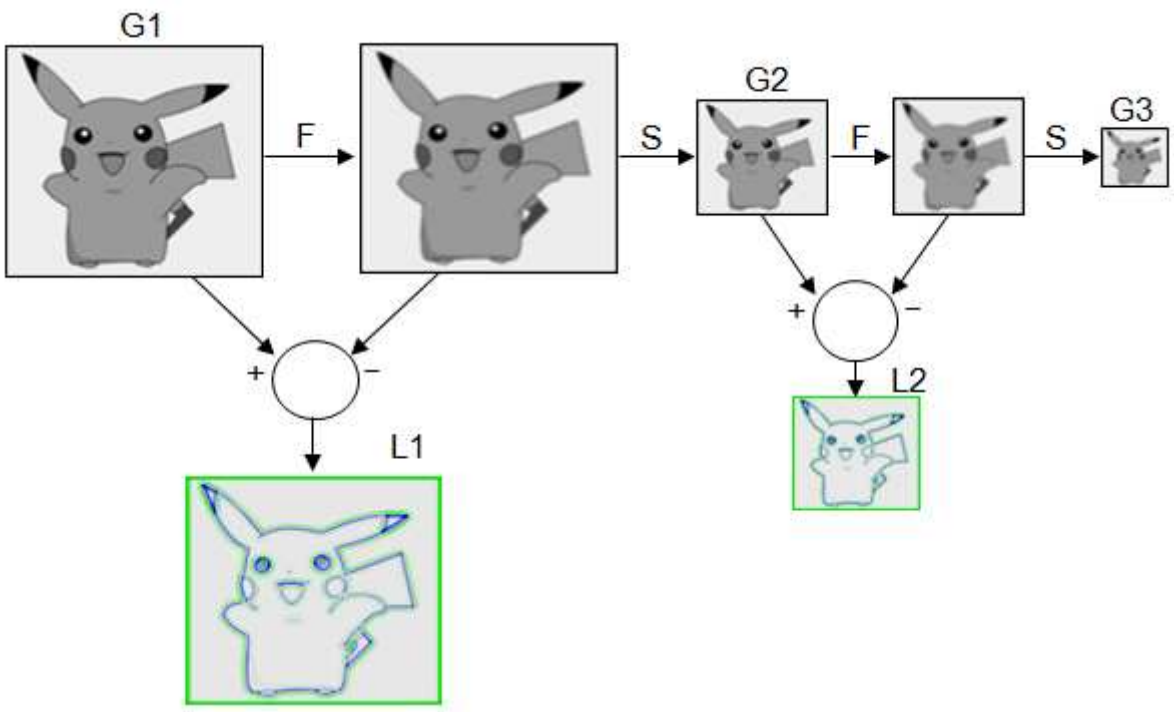

FIGURA 36 : PIRÁMIDE DE LAPLACE

Lo que se guarda son los detalles. Se guarda la resta entre la imagen original y la filtrada.

La siguiente forma de obtener réplicas de la imagen original se produce al reducir el ancho de banda de la imagen original, consiguiendo un efecto borroso sobre la imagen original. Ésta técnica se denomina *Scale-Space* y es la que utiliza el algoritmo de SURF (Figura 37).

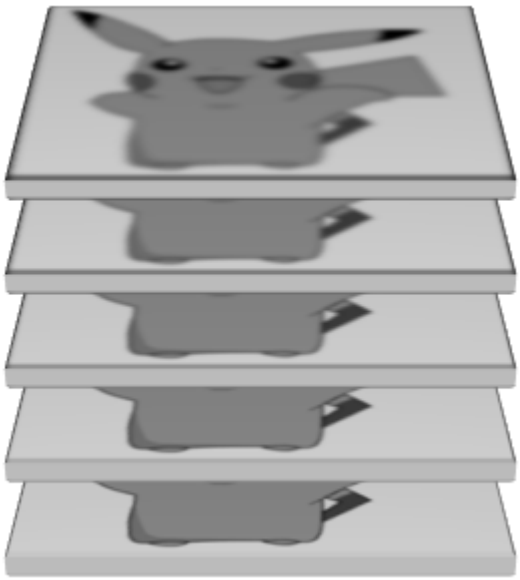

FIGURA 37 : TÉCNICA DE SCALE-SPACE

También se pueden usar variantes de de estas dos opciones como realizar la técnica de scale-space con submuestreo, o realizar las pirámides con otras proporciones.

Una vez realizada la técnica de multi-resolución se debe proceder a seleccionar puntos característicos. El algoritmo de SURF utiliza una variante del determinante hexiano utilizado en el algoritmo de Shi y Tomasi.

$$H(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{xy}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{bmatrix}$$

Ec. 27

Conocemos la expresión

$$\frac{\partial^2 g(\sigma)}{\partial x^2}$$

Ec. 28

Como la derivada de segundo orden del filtro de Gauss. Así L_{xx} (x,σ) es la convolución entre la derivada de segundo orden del filtro de gauss y la imagen en el punto **x**=(*x,y*). Lo mismo ocurre con los otros términos. Estas derivadas se conocen como laplacianas de gausianas.

La ventaja del método de SURF es que en vez de usar gausianas para promediar las zonas de la imagen, se usan cuadrados. Éstos cuadrados son una aproximación que reduce mucho el coste computacional del algoritmo. Ver Figura 38.

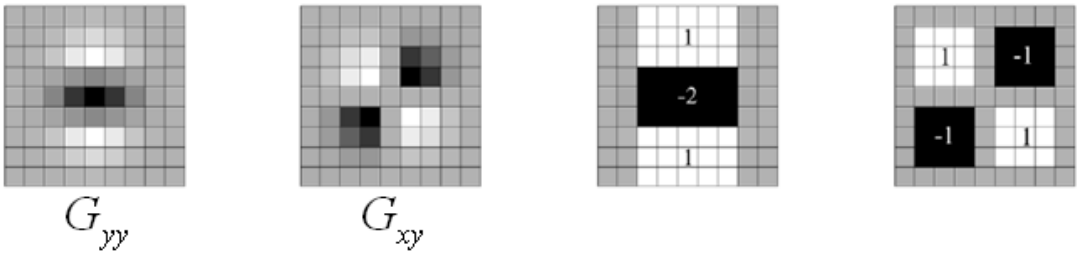

FIGURA 38 : APROXIMACIONES DE LOS FILTROS GAUSIANOS

El gran secreto de la técnica de SURF es la utilización en todos estos procesos de la *imagen integral*. La imagen integral se define de la siguiente manera:

$$S(x,y) = \sum_{x' \le x} \sum_{y' \le y} I(x',y')$$

Ec. 29

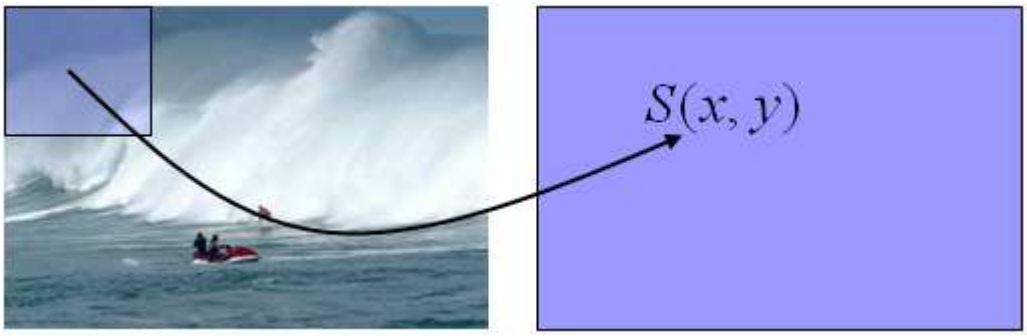

FIGURA 39 : IMAGEN INTEGRAL

Gracias a esta imagen y a la utilización de los cuadrados, la convolución resultante se realiza muy rápida. Tras la convolución se seleccionan los puntos que sean máximos para las diferentes imágenes generadas mediante la multi-resolución.

Una vez seleccionados los puntos característicos se les asigna un descriptor que los hace únicos. Entendámoslo como si fuese la matrícula del punto dado. Con este concepto podemos establecer las correspondencias entre los puntos de dos imágenes; buscando puntos característicos en una imagen, y posteriormente en la otra, y finalmente comparando los descriptores.

Para realizar el descriptor se tienen en cuenta dos aspectos principalmente, la orientación y la descripción de la vecindad. Para la orientación se usan los filtros de Haar, que básicamente nos dan una respuesta en el eje x e y de los puntos de la imagen. En función de esa respuesta se establece la orientación del pixel tratado. Ésta condición en el descriptor hace que los puntos seleccionados sean invariantes también a rotaciones. La descripción de la vecindad se hace mediante un cuadrado centrado en el píxel en cuestión y orientado según la orientación que se ha calculado previamente. El tamaño del cuadrado también depende de los valores de orientación que se hayan calculado.

3.6. IMPLEMENTACIÓN DE LA TÉCNICA DE SURF

Como se comentado anteriormente el procedimiento de selección de puntos homólogos se realiza hallando puntos característicos en una de las imágenes. Se realiza el mismo proceso en la otra imagen. Y posteriormente se comparan los descriptores hallando la distancia euclídea entre ellos. Los descriptores más parejos corresponderán a parejas de puntos homólogos.

Podemos observar en la Figura 40 el desarrollo del algoritmo.

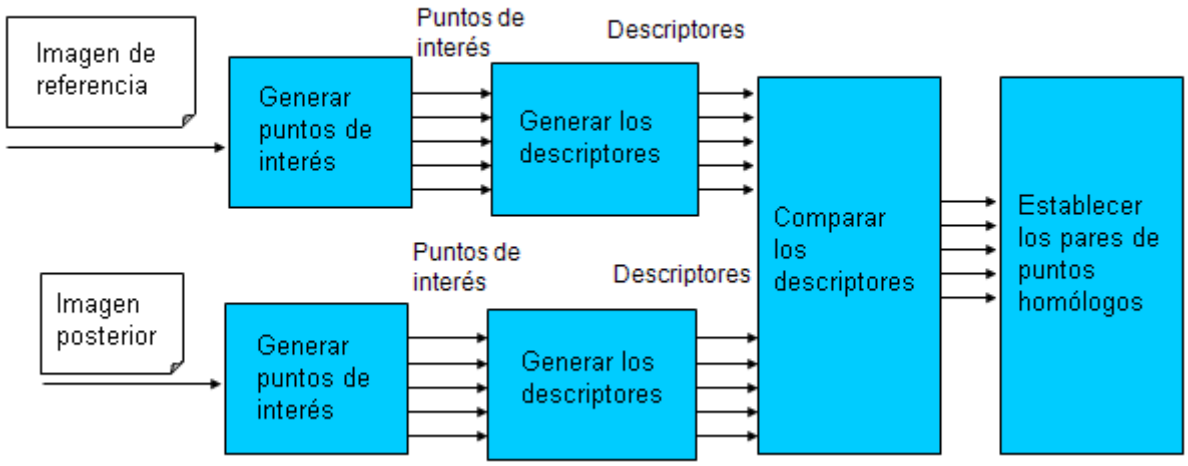

FIGURA 40 : DIAGRAMA DE BLOQUES DEL FUNCIONAMIENTO DEL ALGORITMO DE SURF

Es el único método del proyecto que esta no esta desarrollado en Matlab. Su código es C y se puede descargar libremente de http://www.vision.ee.ethz.ch/~surf/

Tras compilar el código y generar el ejecutable lo único que necesita como argumento son la imagen en cuestión, un umbral (opcional) y el archivo de salida. A medida que el umbral es mayor, el número de puntos escogido es menor. Tras la ejecución se guarda en el archivo de salida una lista de los puntos escogidos. Cada uno de ellos con su descriptor que lo identifica como un punto único. El formato de salida es:

(1 + Longitud del descriptor)

Numero de puntos

x y a b c l des

x y a b c l des

...

x, y = posición del punto característico

a, b, c = [a b; b c] valores correspondientes a la orientación del punto.

l = signo del laplaciano (-1 o 1)

des = descriptor

3.7. RESULTADOS DEL ALGORITMO SURF

El primer caso se prueba con el par de imágenes estéreo que denominamos anteriormente “tsukuba”. El resultado se puede observar en la Figura 41.

FIGURA 41 : "TSUKUBA" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

Podemos observar que la selección delos pares de puntos homólogos la hace de manera correcta para todos los pares. Ahora probaremos la selección de puntos sobre éste mismo par de imágenes pero atendiendo solo a la capa del fondo para ver los resultados que se obtienen.

FIGURA 42 : "TSUKUBA CAPA DEL FONDO" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

Comprobamos que los resultados son correctos, todos los emparejamientos son satisfactorios. A continuación probaremos para las otras tres capas restantes. La correspondiente de la mesa y la cámara de video, la capa del busto y la última capa que corresponde con la lámpara.

FIGURA 43 : "TSUKUBA CAPA DE LA MESA" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

FIGURA 44 : "TSUKUBA CAPA DEL BUSTO" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

FIGURA 45 : "TSUKUBA CAPA DE LA LÁMPARA" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

Podemos comprobar que salvo en la última capa, la de la lámpara, todos los emparejamientos son correctos. Los problemas de la última capa se deben principalmente a la profundidad a la que se encuentran éstos puntos. Recordemos que tras la selección de puntos mediante el algoritmo de SURF se le añade un descriptor a cada punto en el cual se refleja la información del entorno del punto en cuestión. Es decir, cuando se selecciona un punto de la capa de la lámpara se le asigna un descriptor que tiene información de su entorno. Si éste punto esta cerca del borde de la lámpara se le asignará en el descriptor información referente a las capas mas profundas, las que estén detrás de la lámpara. Fijémonos por ejemplo en el punto superior de la lámpara (Figura 46).

FIGURA 46 : AMPLIACIÓN DE LA FIGURA 45

El punto superior pertenece a la capa de la lámpara. Vemos que el punto está mal enlazado respecto a la lámpara. Pero si este enlace lo vemos como un enlace de la imagen del fondo podemos asegurar que es un buen emparejamiento. Esto pasa por asociar al descriptor información de la vecindad del punto. En la imagen derecha primero se selecciona el punto y luego se le asocia información de su contorno. Pero el problema es que éste contorno es principalmente la capa mas profunda, la de los libros. Por eso en la imagen izquierda el descriptor que mas se le parecerá será el que corresponda a algún punto que esté en la capa de los libros, en vez de en la capa de la lámpara. Este error se produce por la diferencia de profundidad entre las dos capas que estamos tratando. Comprobamos que aunque ampliemos el número de puntos característicos que encuentra la técnica de SURF, aunque el resultado mejore, no es satisfactorio.

FIGURA 47 : "TSUKUBA CAPA DE LA LÁMPARA CON MAYOR SELECCIÓN DE PUNTOS CARACTERÍSTICOS" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

Una posible solución es trabajar con imágenes de mayor resolución. Si operamos con imágenes mayores el número de puntos que detectará el algoritmo SURF será mayor, así en la posterior comparación de los correspondientes descriptores se podrán encontrar mejores emparejamientos. Para probar esta posible solución trabajaremos con el par estéreo que denominamos anteriormente como “cartas”. Estas imágenes fueron tomadas con una resolución inicial de 1920x1068 y posteriormente se redujeron para evitar el coste computacional asociado al tamaño. Probaremos primero con las imágenes reducidas para comprobar que se siguen produciendo estos errores de correspondencia y posteriormente probaremos con las imágenes originales para observar si se solucionan los problemas que acabamos de ver.

FIGURA 48 : "CARTAS, CAPA DE LA CARTA DE PÓKER, SELECCIÓN DE PUNTOS CARACTERÍSTICOS" SELECCIÓN DE PUNTOS HOMÓLOGOS MEDIANTE SURF.

La imagen de arriba corresponde con la cámara izquierda y la de abajo con la cámara de la derecha. Parece que los emparejamientos son correctos pero si ampliamos la imagen observaremos que no todas las correspondencias son buenas.

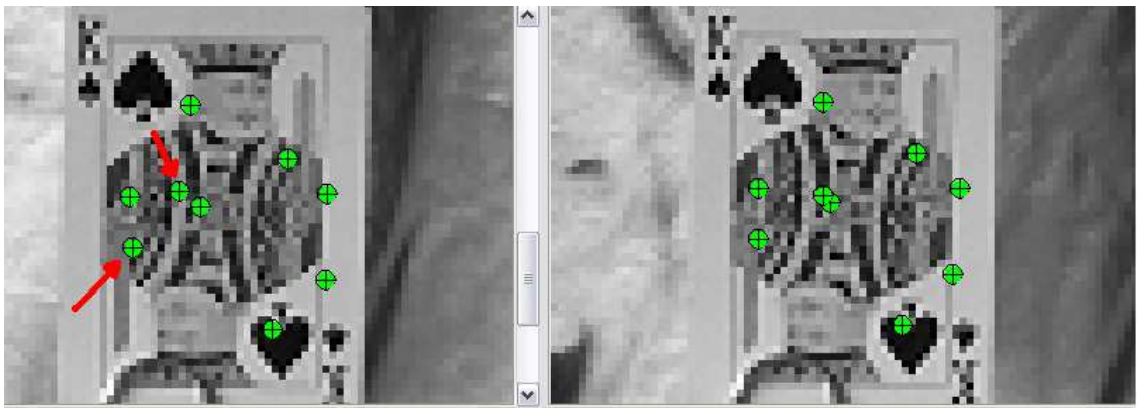

FIGURA 49 : ERRORES DE EMPAREJAMIENTO EN EL PAR "CARTAS" PARA LA CAPA DE LA CARTA DE PÓKER

Los errores no son del todo malos pero no son correctos. Si observamos la Figura 50 podemos ver que ahí si que los emparejamientos son peores. Observar el punto lateral izquierdo en ambas cartas.

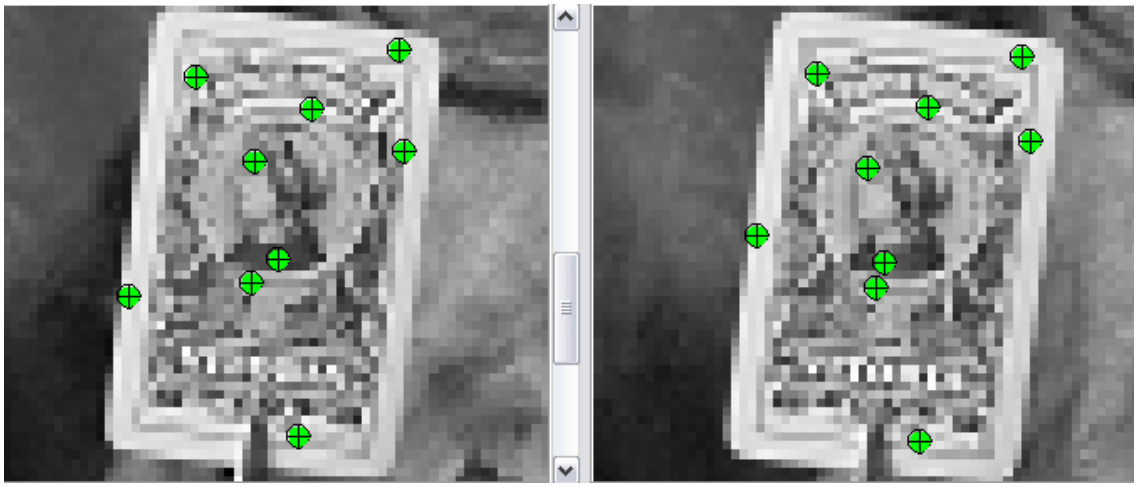

FIGURA 50 : ERRORES DE EMPAREJAMIENTO EN EL PAR "CARTAS" PARA LA CAPA DE LA CARTA ESPAÑOLA

Probaremos ahora con las imágenes sin reducir. Podemos ver en la Figura 51 las ampliaciones de la capa de la carta de póker para las imágenes originales del par “cartas”.

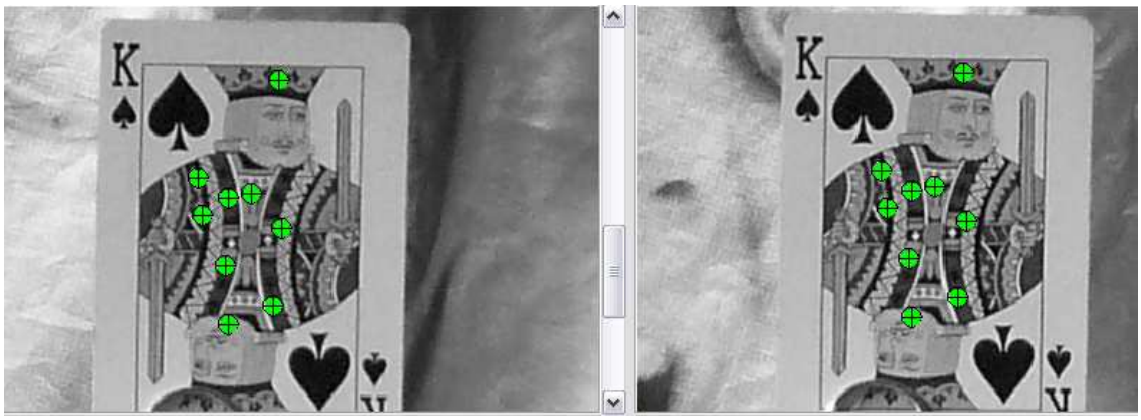

FIGURA 51 : EMPAREJAMIENTOS CORRECTOS EN EL PAR "CARTAS" PARA LAS IMÁGENES ORIGINALES (CAPA DE LA CARTA DE PÓKER)

Podemos comprobar que todos los emparejamientos son perfectos. Si comprobamos las siguientes capas observaremos que los resultados son óptimos.

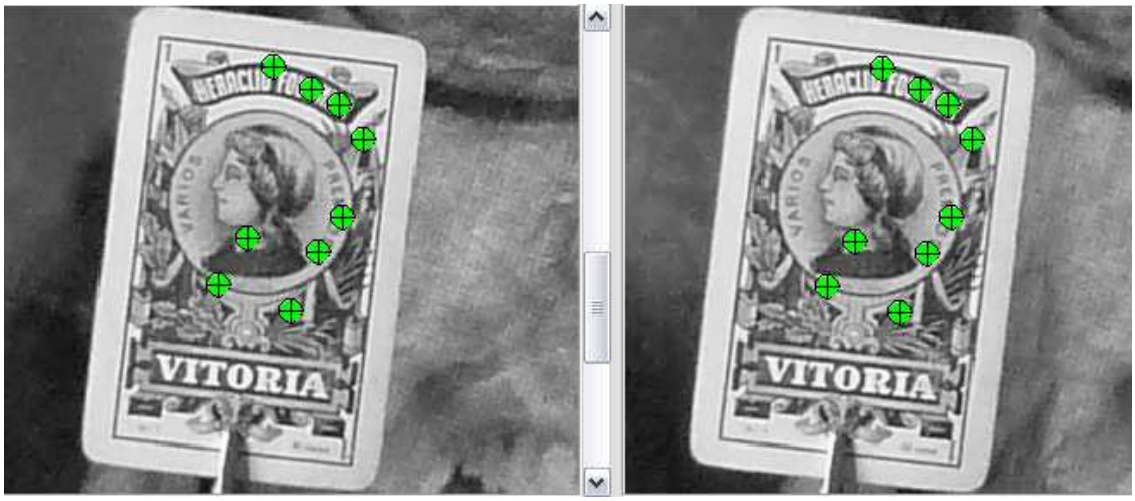

FIGURA 52 : EMPAREJAMIENTOS CORRECTOS EN EL PAR "CARTAS" PARA LAS IMÁGENES ORIGINALES (CAPA DE LA CARTA ESPAÑOLA)

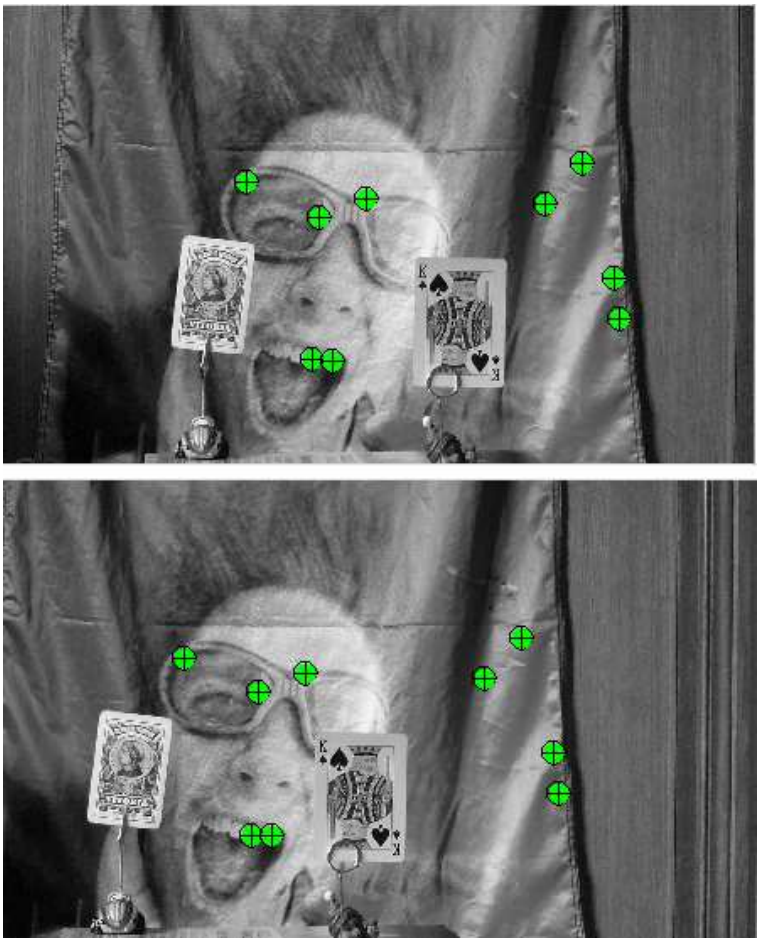

FIGURA 53 : EMPAREJAMIENTOS CORRECTOS EN EL PAR "CARTAS" PARA LAS IMÁGENES ORIGINALES (CAPA DEL FONDO)

3.8. CONCLUSIONES DEL ALGORITMO SURF

Tras observar los resultados del algoritmo SURF podemos concluir que se ajusta perfectamente a las condiciones de uso que necesitamos. Puede seleccionar puntos por capas sin necesidad de dividir la imagen en cuadrantes. La única salvedad sería utilizar imágenes con buena resolución. Además el tiempo de ejecución en comparación con el algoritmo de Shi y Tomasi es mínimo. Otra de las ventajas es la invarianza ante cambios de escala y ante rotaciones. Estos requisitos no son necesarios para el desarrollo de nuestro proyecto pero debemos de tenerlos en cuenta pues los pares de puntos correspondientes serán seleccionados de manera más robusta. Por todo ello utilizaremos la técnica de SURF para establecer los pares de puntos homólogos.

4. OBTENCIÓN DE MAPAS DE PROFUNDIDAD A PARTIR DE DOS IMÁGENES

4.1. ALGORITMO ESTUDIADO

Es éste cuarto capítulo se describirá el algoritmo utilizado para la generación del mapa de profundidad de la escena grabada. Es un algoritmo cooperativo (o de relajación) que fue expuesto por C. Lawrence Zitnick, Takeo Kanade en 1999 [28]. También se caracteriza por la detección de puntos ocluidos.

Se eligió éste algoritmo por sencillez y por su no demasiado elevado coste computacional. Los algoritmos basados en programación dinámica o corte de grafos requieren un gran coste computacional y son más complejos de desarrollar.

A continuación procedemos a describirlo.

4.1.1. ALGORITMO DE DISPARIDAD

Es un método basado en utilización de una ventana para detectar disparidades mediante algún tipo de correlación. El tamaño de dicha ventana no es una decisión trivial, depende de la textura, formas, tamaños, etc. Para imágenes bien texturizadas se puede aplicar un tamaño de ventana pequeño, si no es así el tamaño de la ventana lo deberíamos aumentar.

El algoritmo utilizado proporciona un mapa denso de la disparidad de la escena, es decir, proporciona información de toda la escena de forma general (no muestra la disparidad de un único par de puntos por ejemplo). Además detecta explícitamente puntos ocluidos. Se deben asumir dos condiciones de funcionamiento:

Unicidad: Se define como que para cada punto de una imagen referencia existe solamente un único punto correspondiente en la imagen posterior. La condición de unicidad se cumple siempre que en la escena no existan objetos semitransparentes (varios puntos sobre la misma recta proyección generan un mismo punto imagen en el sensor). Cuando no hay oclusión la relación de correspondencia es bidireccional. Mientras que si existe oclusión, hay puntos que no tienen correspondencias.

Continuidad: Los puntos proyectados sobre la imagen pertenecen a las superficies de los objetos de la escena. Estas superficies se asumen continuas presentando discontinuidades únicamente en la separación de los objetos (principio de cohesión de la materia). Esta continuidad de las superficies se traduce en una continuidad en el mapa de profundidades. Como se ha visto, la profundidad y la disparidad están estrechamente relacionadas, por lo que esta restricción se impone como una continuidad de las disparidades en la escena. Ésta condición está ligada al proceso de relajación.

Como se introdujo anteriormente se usará el cubo de correlación para establecer las correspondencias entre los valores de disparidad y los puntos de las imágenes. Cada valor del cubo representa la posición de los puntos de la imagen de referencia y su valor es el valor de disparidad respecto de la imagen posterior. Es decir, si tenemos el punto (x1,y1) de la imagen de referencia, en la imagen posterior tendremos el mismo punto en la coordenada (x2,y2) donde x2=x1+d1 (Se asumen imágenes rectificadas). Así en el cubo de correlación deberíamos tener un valor alto en la posición (x1,y1,d1). Mostramos el cubo de correlación o espacio de disparidad en la Figura 54.

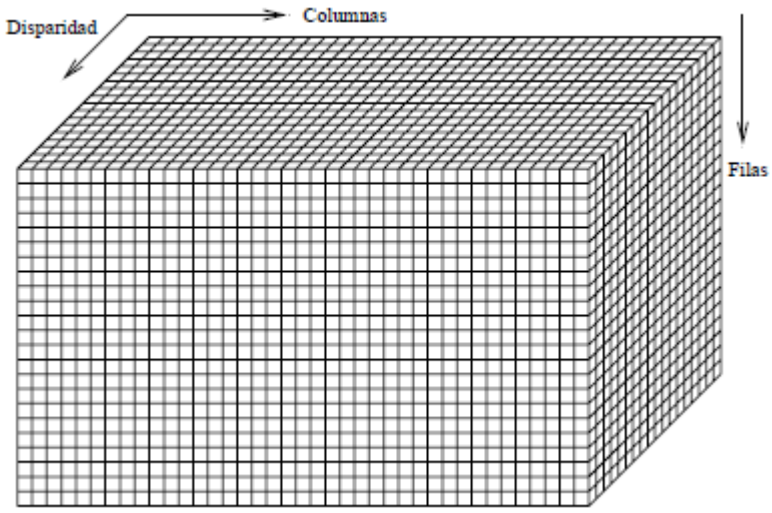

FIGURA 54 : ESPACIO DE DISPARIDAD

El cubo de correlación tiene 3 ejes:

Fila → r

Columna →c

Fondo →disparidad d

Asumimos que las imágenes han sido rectificadas (sin perder generalidad), así un elemento (r,c,d) se proyecta a la imagen izquierda como (r,c) y a la derecha como (r,c+d).

Primeramente se establece un emparejamiento entre puntos mediante la técnica de correlación. Este primer paso es una primera aproximación al mapa de disparidad. A continuación se utiliza una función de actualización que proporciona

valores continuos y únicos. Para ello se establece algún promediado entre los valores que hay entre los puntos vecinos y los valores de *inhibición* (veremos a continuación su significado) a través de líneas similares. A medida que el algoritmo converge se van detectando los puntos ocluidos.

Para entender el algoritmo nos detendremos en la siguiente imagen.

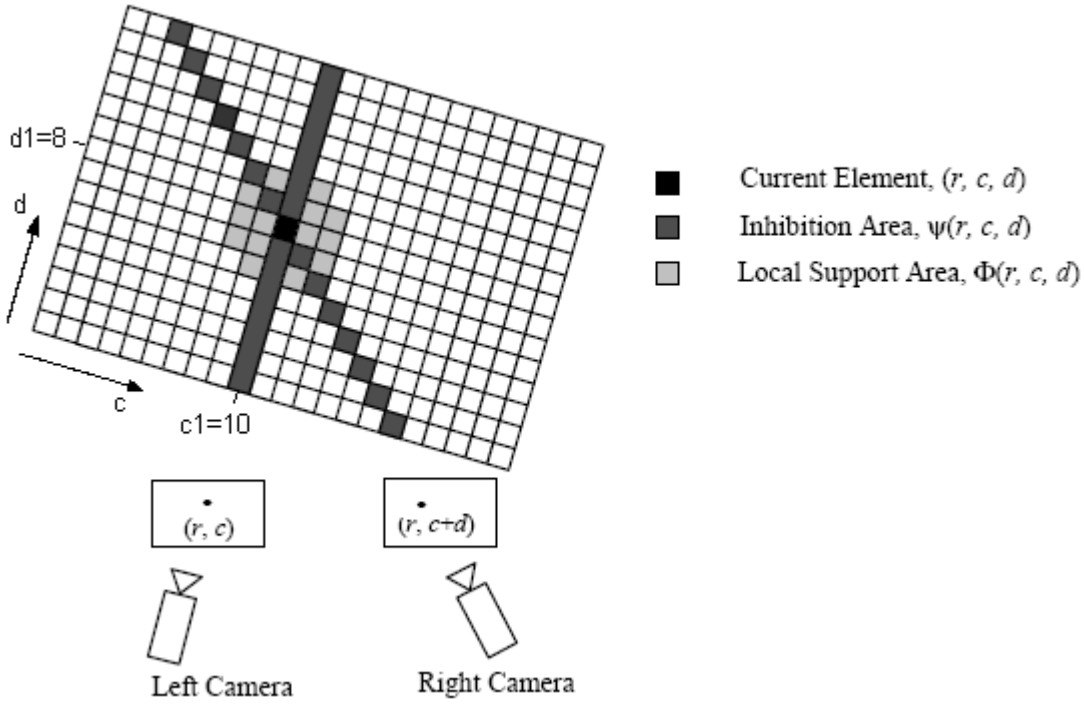

FIGURA 55 : ILUSTRACIÓN DE LAS ÁREAS DE INHIBICIÓN, Y DE SOPORTE DENTRO DEL CUBO DE CORRELACIÓN. PARA UN NÚMERO DE FILA FIJO.

Corresponde a una de las capas horizontales del cubo de correlación, es decir, para una fila determinada (por ejemplo r=r1). Así el punto (c1,r1) del cubo de correlación, corresponde con la posición en la imagen izquierda del píxel (x1,y1). Para x1=c1 y y1=r1. El rango de d es el rango de disparidad que hayamos definido al inicio del algoritmo. Según éste ejemplo d=1 corresponde con la disparidad mínima y d=14 se corresponde con la disparidad máxima. Que d=1 corresponda con la disparidad mínima no quiere decir que la disparidad mínima sea 1. Puede ser por ejemplo 4 ó -2. Lo que si importa es que el rango de d=1 hasta d=14 tiene el mismo tamaño que el rango que va desde la disparidad mínima a la máxima. Por ejemplo podríamos tener configurado un rango de disparidades para éste ejemplo de -2 hasta 11 donde -2 correspondería con la posición en el cubo de correlación de d=1 y 11 sería d=14.

Volvemos a la Figura 55 y con la posición de (r1,c1). Los valores de d para esa posición corresponden con valores de emparejamiento para cada d. Por ejemplo el valor de (r1,c1,d=1) corresponde con el valor de emparejar por correlación el valor de la imagen izquierda (x1,y1) con el valor de la imagen derecha (x1 - 2,y1) (manteniendo el rango de ejemplo del párrafo anterior). Podemos ver que según ese emparejamiento el valor que le corresponde la punto (r1,c1) es el que corresponde con d=8, es decir, una disparidad de 5 (si mantenemos el mismo rango del párrafo anterior). Viendo los

resultados respecto a las imágenes podemos decir que el punto de la imagen izquierda (x1,y1) tiene su correspondiente punto en la imagen derecha a una disparidad de 5 píxeles (x2=x1+d; para d=5).

Para los demás valores de d el resultado de emparejamiento debe ser menor puesto que corresponderían con puntos distintos del elegido. Según la condición de continuidad podemos asegurar que emparejamientos vecinos deben tener valores parecidos de emparejamiento. Por ello todos los puntos que corresponden con el área de inhibición deben contener valores pequeños y los que pertenecen al área de soporte (o excitatoria) deben contener valores similares al del punto actual (r1,c1,d=8). Idealmente el área de soporte solo correspondería con dos puntos a cada lado de punto actual, serían los vecinos reales. Pero como no se sabe de antemano estos emparejamientos se determina como se muestra en el ejemplo. Lo único que se descarta son los puntos que pertenecen al área de inhibición que sabemos que son emparejamientos incorrectos. Extendiendo el concepto de continuidad sobre las filas debemos construir un área de soporte tridimensional, donde no se contará con las áreas de inhibición de cada capa correspondiente a cada fila.

Continuamos con la explicación del algoritmo.

L_n(r,c,d) denota el valor del emparejamiento asignado al elemento (r,c,d) en la iteración n.

Como comentábamos anteriormente primeramente se debe establecer algún tipo de correspondencia inicial entre los puntos de las dos imágenes. Esta corresponde con L_0(r,c,d). Puede ser computado inicialmente de la siguiente manera:

$$L_0 = \delta(Im_{left}(r,c), Im_{right}(r,c+d)) \qquad \text{Ec. 30}$$

Donde δ es una función de comparación sobre las imágenes tal como la correlación normalizada, la suma absoluta de diferencias o la diferencia de cuadrados. En nuestra implementación se ha desarrollado la comparación por correlación normalizada y también la comparación por suma absoluta de diferencias (SAD).

Este primer paso debería de dar valores altos para puntos correspondientes, pero debido a las oclusiones también pueden aparecer valores altos para emparejamientos erróneos.

La siguiente presunción implica que los emparejamientos vecinos tienen valores similares. Para ello necesitamos escoger un tamaño de ventana que no es sabido de antemano. Como comentábamos el área de soporte será definida como un espacio tridimensional:

S_n (r,c,d) es el valor de coste del área de soporte centrada en el punto (r,c,d). Por ejemplo la suma de los valores de nuestra 3-D área local:

$$S_n(r,c,d) = \sum_{(r',c',d')\in\theta} L_n(r+r', c+c', d+d') \qquad \text{Ec. 31}$$

Donde θ es nuestra área 3-D.

Ahora aplicamos la idea de puntos únicos lo que implica que solo puede existir un emparejamiento dentro del conjunto de elementos que proyectan a un mismo píxel en una imagen. Este conjunto de píxeles es conocido como el área de inhibición, tal y como se muestra en la Figura 55 →φ . El área de inhibición se puede entender como el conjunto de puntos que se proyectan sobre el punto (x1,y1) de la imagen izquierda o sobre el punto (x1+d,y1) de la imagen derecha. Debemos calcular el valor del conjunto de éstos puntos para el espacio tridimensional sobre el que se evalúe el área de soporte.

$R_n(r,c,d)$ denota la cantidad de inhibición que $S_n(r,c,d)$ recibe por los elementos de $\varphi_n(r,c,d)$:

$$R_n(r,c,d) = \left(\frac{S_n(r,c,d)}{\sum_{(r'',c'',d'')\in\varphi} L_n(r'',c'',d'')}\right)^{\alpha} \qquad \text{Ec. 32}$$

El exponente α controla la cantidad de inhibición en cada iteración. Para garantizar la unicidad el valor de α debe ser mas grande que 1.

Puesto que hemos reducido los valores del área de soporte según los valores del área de inhibición, se procede aumentar los posibles valores correctamente emparejados mediante la primera iteración que habíamos realizado: $L_0(r,c,d)$ recordemos que tras esta iteración los emparejamientos correctos deberían tener valores altos. Así obtenemos $T_n(r,c,d)$ que denota el valor condicionado de $R_n(r,c,d)$ por $L_0(r,c,d)$:

$$T_n(r,c,d) = L_0(r,c,d) * R_n(r,c,d) \qquad \text{Ec. 33}$$

Así nuestra función de actualización es:

$$L_{n+1}(r,c,d) = L_0(r,c,d) * \left(\frac{S_n(r,c,d)}{\sum_{(r'',c'',d'')\in\varphi} L_n(r'',c'',d'')}\right)^{\alpha} \qquad \text{Ec. 34}$$

4.1.2. DETECCIÓN EXPLÍCITA DE ÁREAS OCLUIDAS

La detección de puntos ocluidos es realizable imponiendo la restricción de unicidad que asumimos anteriormente. Debido a la condición de unicidad los puntos que tengan valores pequeños serán ocluidos y estarán en áreas cercanas. Observemos la Figura 56 para comprender los dos principales casos de puntos ocluidos.

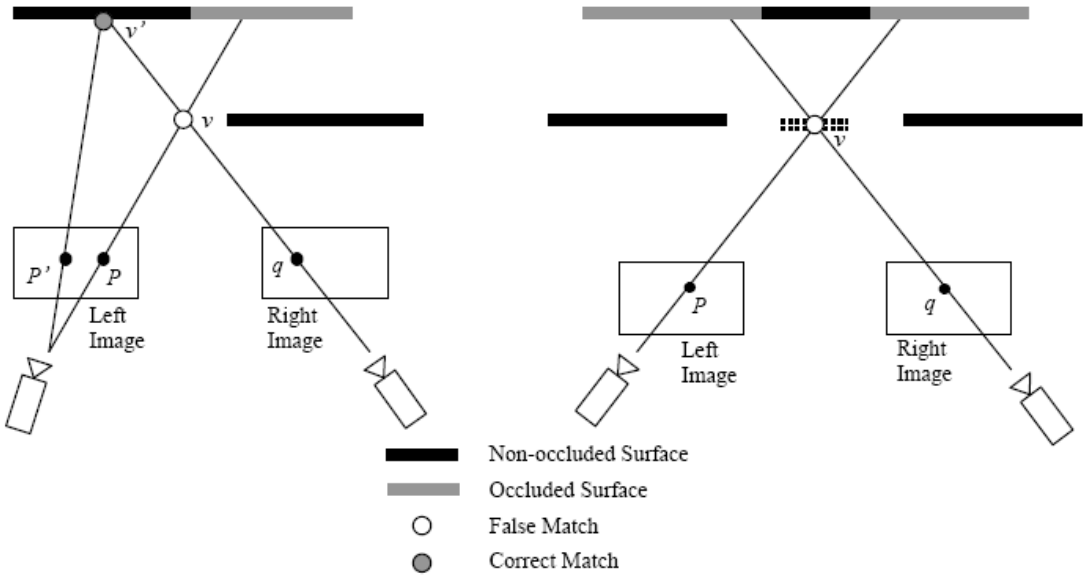

FIGURA 56 : PRESENTACIÓN DE LOS DOS PRINCIPALES CASOS REFERENTES A LAS OCLUSIONES

En el ejemplo de la izquierda vemos como el punto *q* de la imagen de la derecha le corresponde el punto *P*' de la imagen izquierda. Pero observamos que el punto *P* también se proyecta sobre el punto *q*. Gracias a la condición de unicidad, que impusimos al ponderar el área de soporte mediante el área de inhibición, lo normal es que el emparejamiento de *q* con *P*' tenga un valor mayor y sea el seleccionado. El enlace con *P* sería descartado y obtendría valores bajos. Al final el punto *P'* sería considerado un punto ocluido.

En el ejemplo de la derecha se muestra el ejemplo menos satisfactorio que se puede dar. En éste caso el punto *q* de la imagen derecha no tiene correspondencia con ninguno de la imagen izquierda, pero al proyectarse sobre él puntos de la imagen izquierda que también son ocluidos (y por lo tanto no tendrán emparejamientos con valores altos) alguno de los enlaces entre este conjunto de puntos será dado como válido, en éste caso el punto *P*. Esto se debe a la presencia de dos zonas ocluidas en cada cámara donde las proyecciones de sus puntos se cruzan. Éste caso sería prudente evitarlo en la medida de lo posible.

4.1.3. RESUMEN DEL ALGORITMO

1. Crear una matriz (r,c,d) :
   a. R: altura de la imagen de referencia.
   b. C: anchura de la imagen de referencia.
   c. D: rango de disparidades.
2. Crear Lo mediante la correlación normalizada o la diferencia de cuadrados.
3. Ir calculando Ln con la fórmula propuesta hasta la convergencia de los valores de emparejamiento.
4. Para cada píxel (r,c) encontrar el máximo valor de (r,c,d).
5. Si el valor supera un umbral se clasifica como la disparidad del píxel dado. Si no supera el umbral se clasifica como punto ocluido.

4.2. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

El funcionamiento de este algoritmo necesita de unos parámetros iniciales no comunes a todas las imágenes. Las variables mas importantes son: el número de iteraciones (N_Iteraciones), el valor máximo (d_Max) y mínimo de disparidad (d_Min) , el valor del radio para la correlación en la primera iteración (radio_Lo), el valor del radio para las demás iteraciones (radio) y el valor de la profundidad en la búsqueda de las demás iteraciones (radio_D). Además se puede configurar la opción de correspondencia para que aplique la fórmula de la correlación o la de suma absoluta de diferencias SAD.

Los autores del algoritmo aconsejan ciertos valores en algunos de estos parámetros:

N_Iteraciones entre 5 y 15

radio_Lo entre 1 y 2

radio entre 1 y 3

radio_D igual a 1

Todos los mapas de profundidad mostrados son relativos a la imagen de la derecha.

Las primeras pruebas las realizamos con el par de imágenes que denominaremos “cartas” de tamaño 207x384. Estas imágenes están tomadas por una misma cámara en dos instantes diferentes de tiempo y desplazadas.

FIGURA 57 : LA IMAGEN IZQUIERDA CORRESPONDE CON LA FOTOGRAFÍA DE LA CÁMARA IZQUIERDA Y LO MISMO PARA LA DERECHA.

Adjuntamos también el mapa de disparidad real del par estéreo asociado a la imagen derecha. Dicho mapa está realizado “a mano”.

FIGURA 58 : "CARTAS" MAPA DE DISPARIDAD REAL.

La primera prueba se hace con un rango de disparidad de d_min = 38 a d_max= 70, el valor de los radios es radio_lo=2, radio = 3, radio_d =1. El número de iteraciones es 1.Se usa la correlación en vez de SAD.

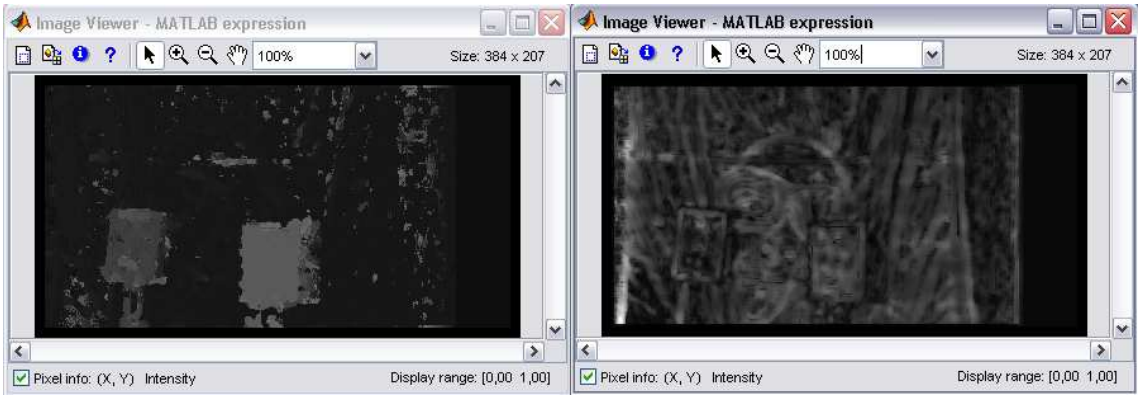

FIGURA 59 : “CARTAS” ITERACIONES 1

La imagen de la izquierda corresponde con el mapa de profundidad. La imagen de la derecha contiene los valores de los emparejamientos, servirá para deducir que

puntos son ocluidos o no. Si un emparejamiento tiene valor bajo (es negro) se puede intuir que es un punto ocluido.

En el mapa de profundidad podemos ver como la zona correspondiente a la carta de póker es la mas clara, eso quiere decir que es la mas cercana a la cámara. Y el fondo es lo más oscuro, es decir lo más lejano.

Ahora veremos que ocurre para 5 iteraciones manteniendo los demás parámetros igual.

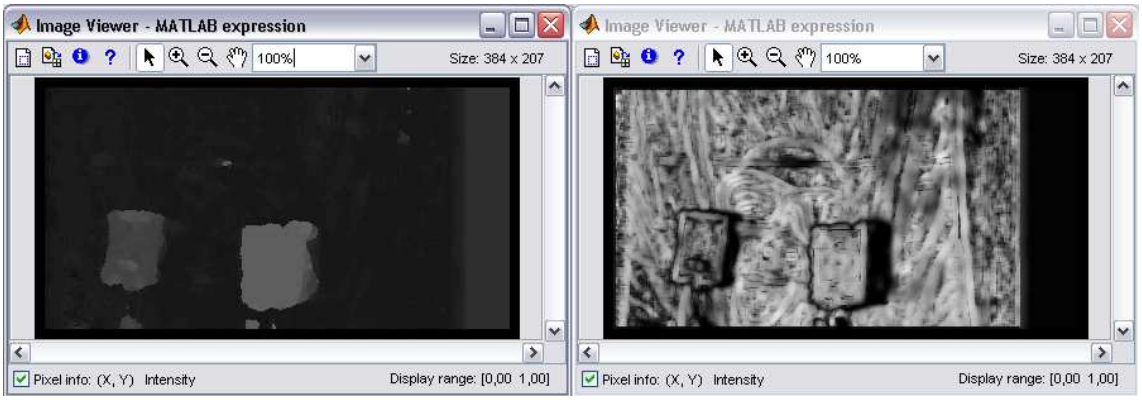

FIGURA 60 : “CARTAS” ITERACIONES 5

Se observa como al aumentar el número de iteraciones la definición es mayor. En la imagen de la derecha se observa como los puntos de la derecha del todo y los de la derecha de las cartas están oscuros. Eso nos va indicando las posibles partes ocluidas.

Ahora vamos a incrementar el número de iteraciones a 10, 20 y 40 manteniendo los demás parámetros como están.

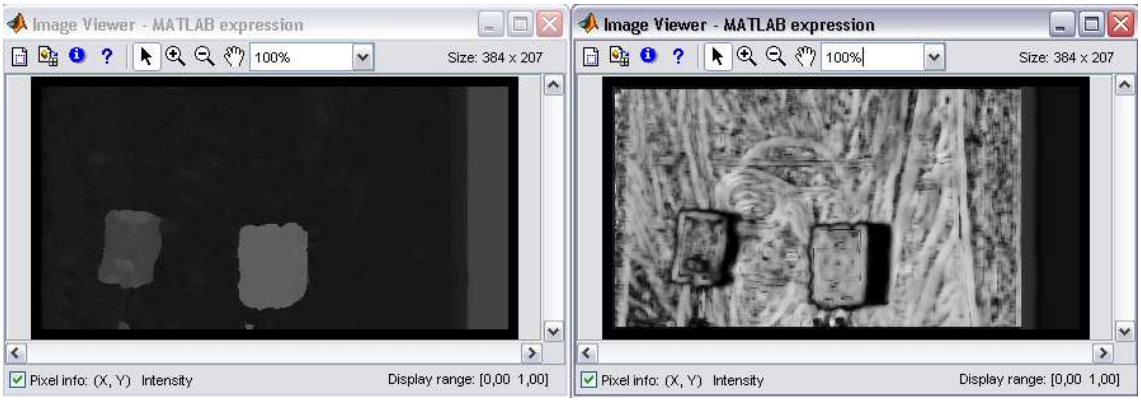

FIGURA 61 : "CARTAS" , N_ITERACIONES=10

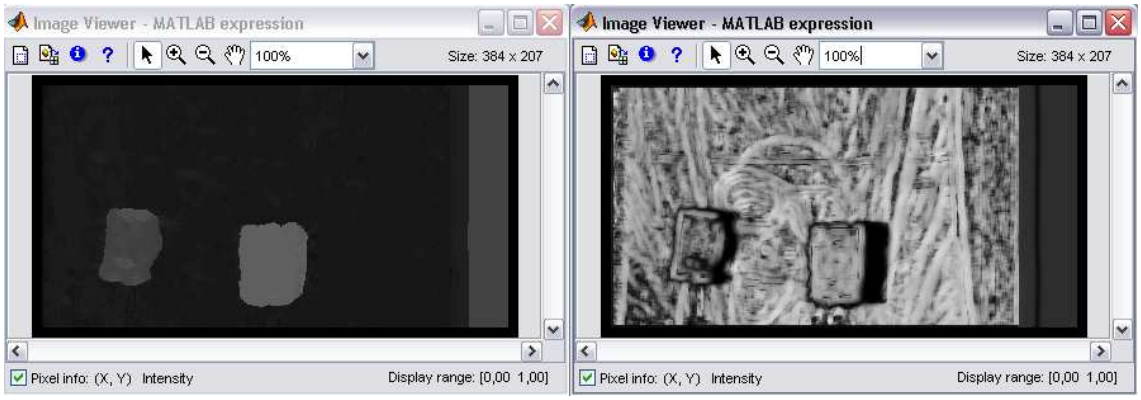

FIGURA 62 : "CARTAS", N_ITERACIONES=20

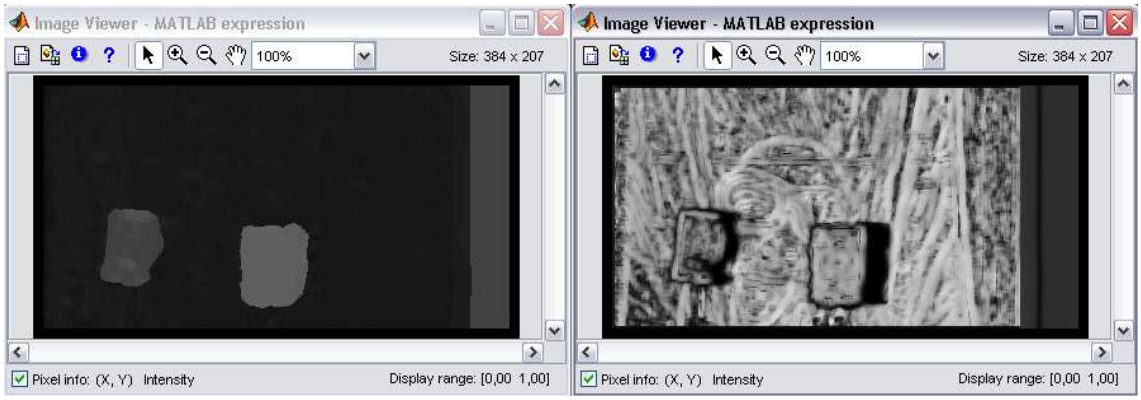

FIGURA 63 : "CARTAS", N_ITERACIONES=40

Se observa como a partir de 10 iteraciones las diferencias no son muy relevantes. Como nos recomiendan los autores un número entre 5 y 15 de iteraciones está bien. Escogeremos el valor de 10 para las demás pruebas.

Como se comentaba anteriormente en la imagen de la derecha los valores mas oscuros corresponden con los puntos ocluidos, los valores de la derecha de las cartas y los de la derecha de la imagen.

Ahora probaremos a reducir el radio de búsqueda a radio_lo = 1, radio = 2 y radio_d = 1. Con 10 iteraciones y usando correlación.

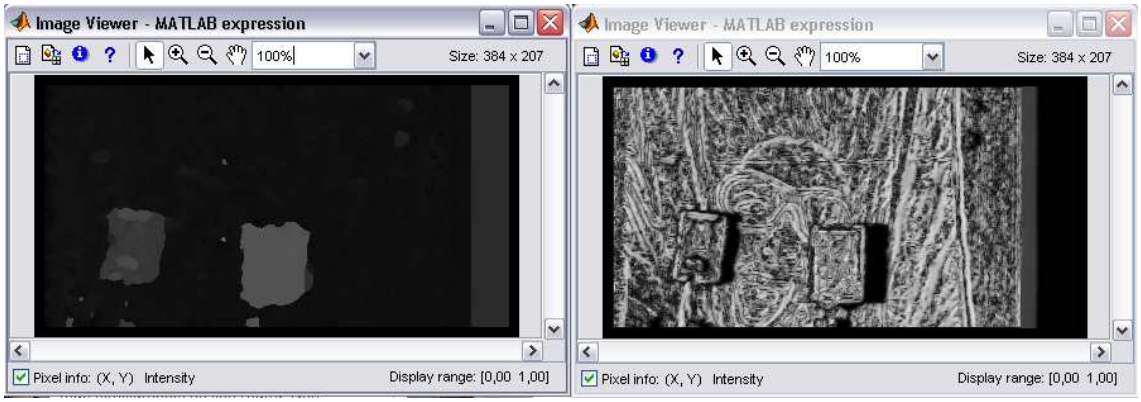

FIGURA 64 : "CARTAS", CON RADIO_LO =1, RADIO =2, RADIO_D=1, ITERACIONES=10

En comparación con la Figura 61 se observan algunos errores más en la imagen que muestra la profundidad de la escena.

Probaremos a usar el método SAD con los parámetros de la Figura 61 que a la vista son los mejores resultados obtenidos.

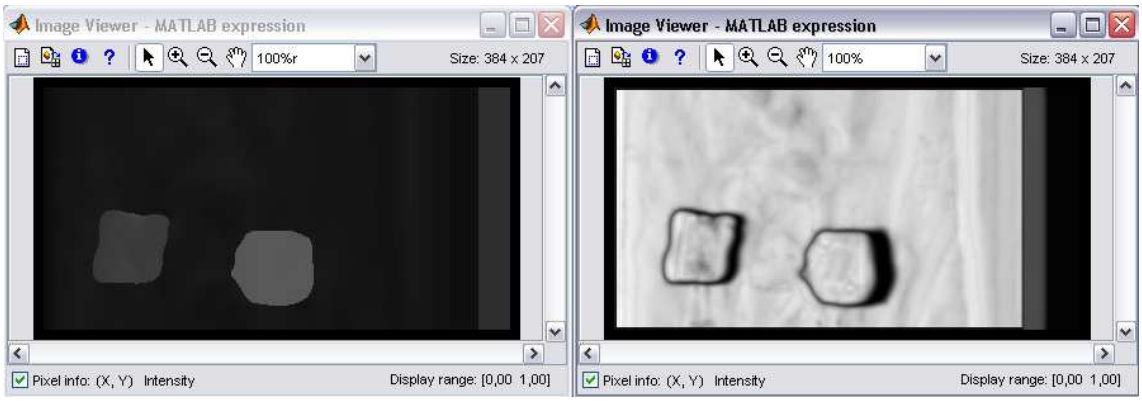

FIGURA 65 : "CARTAS", CON LA TÉCNICA DE SAD

Observamos que el resultado es bastante satisfactorio.

Ahora haremos las pruebas sobre el par de imágenes llamado “tsukuba”. Son un par de imágenes estéreo extraídas de internet de 288x384. Tienen muchas profundidades no diferenciadas claramente.

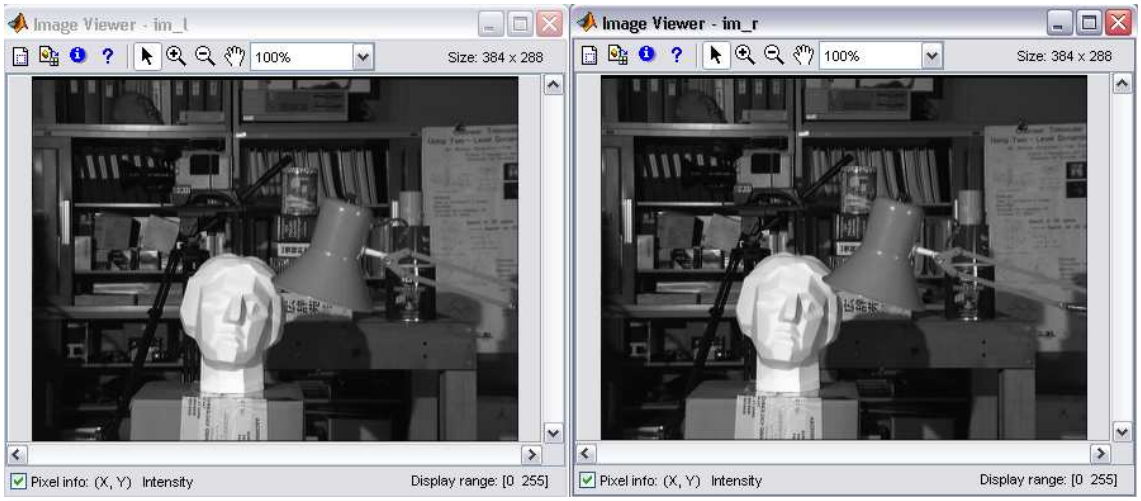

FIGURA 66 : "TSUKUBA" IMAGEN IZQUIERDA E IMAGEN DERECHA

Adjuntamos también el mapa de disparidad real asociado a la imagen izquierda.

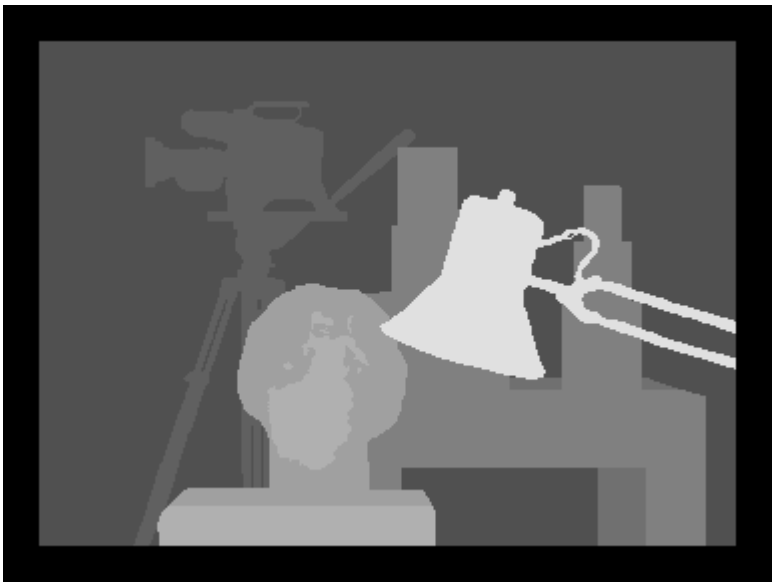

FIGURA 67 : "TSUKUBA" MAPA DE PROFUNDIDAD ASOCIADO A LA IMAGEN IZQUIERDA

Empezaremos las pruebas con los parámetros radio_lo=2, radio = 3, radio_d = 1, iteraciones = 10.

El rango de disparidad estimado es de d_min = 0, d_max=20.

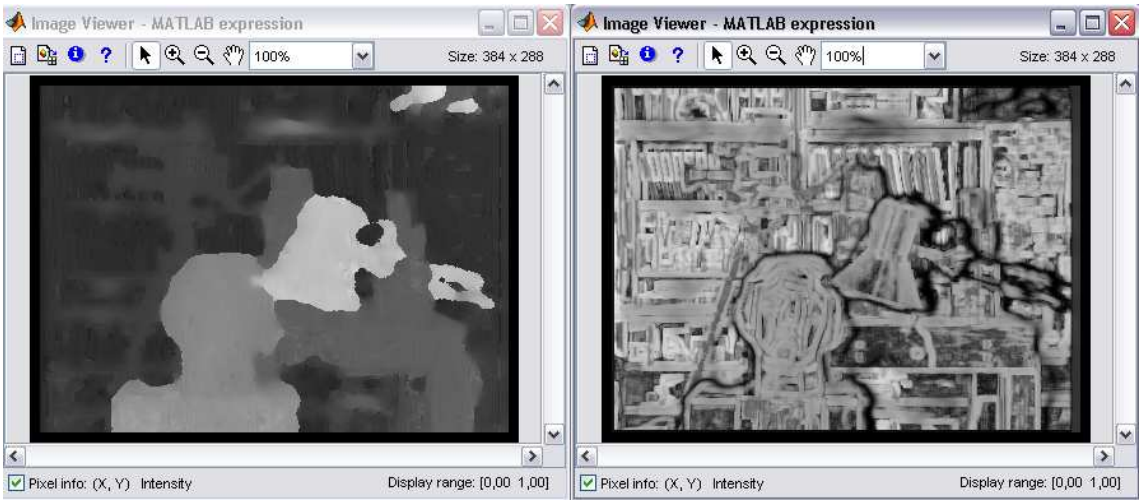

FIGURA 68 : “TSUKUBA” ITERACIONES 10.

Se observan como la primera capa es la que corresponde con la lámpara, la siguiente con el busto y después la mesa y la cámara de vídeo hasta la última capa que es el fondo. En la imagen de la derecha se observa también los posibles puntos ocluidos marcados como puntos oscuros.

Vamos a probar con la técnica de SAD y los mismos parámetros.

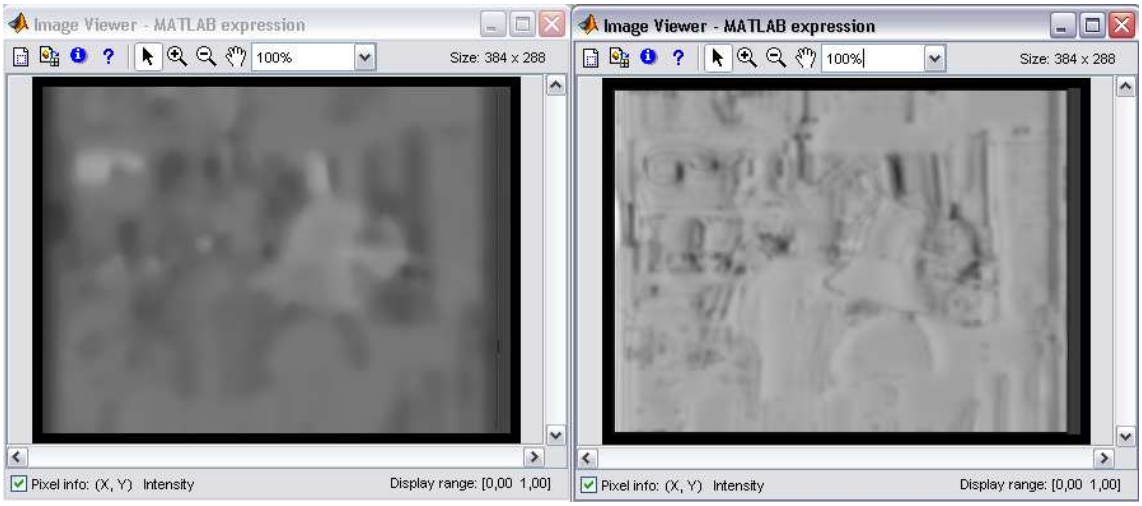

FIGURA 69 : “TSUKUBA” CON SAD.

Como puede apreciarse esta vez la técnica de SAD no ha funcionado muy bien.

Como última prueba con éste par estéreo utilizaremos la técnica de correlación pero disminuyendo el radio a radio_lo= 1, radio=2, radio_d =1. Seguimos con 10 iteraciones.

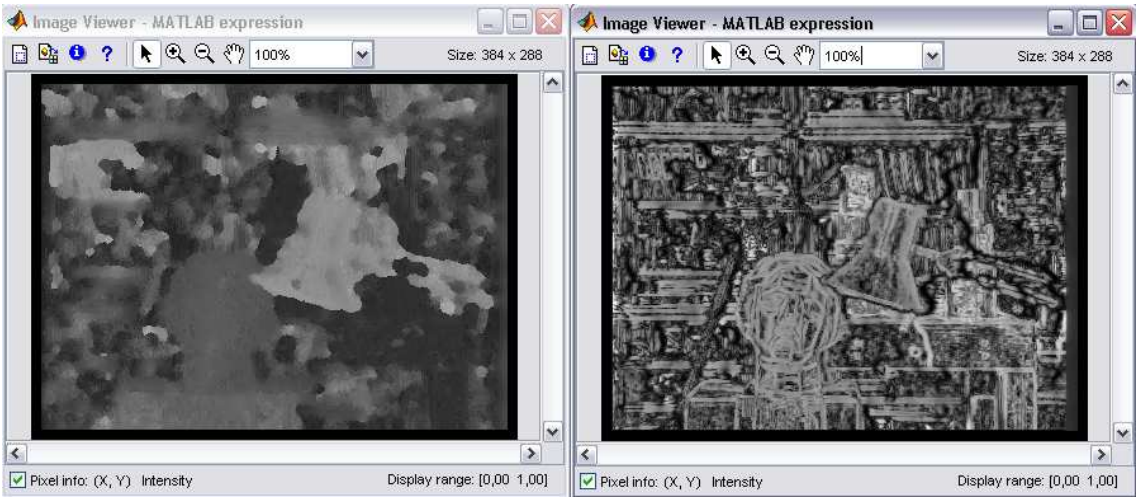

FIGURA 70 : “TSUKUBA” RADIO_LO= 1, RADIO=2, RADIO_D =1

Observamos que tampoco mejoran los resultados. Se podría pensar que aumentando el número de iteraciones mejoraría el resultado. Probamos incrementando las iteraciones hasta 15.

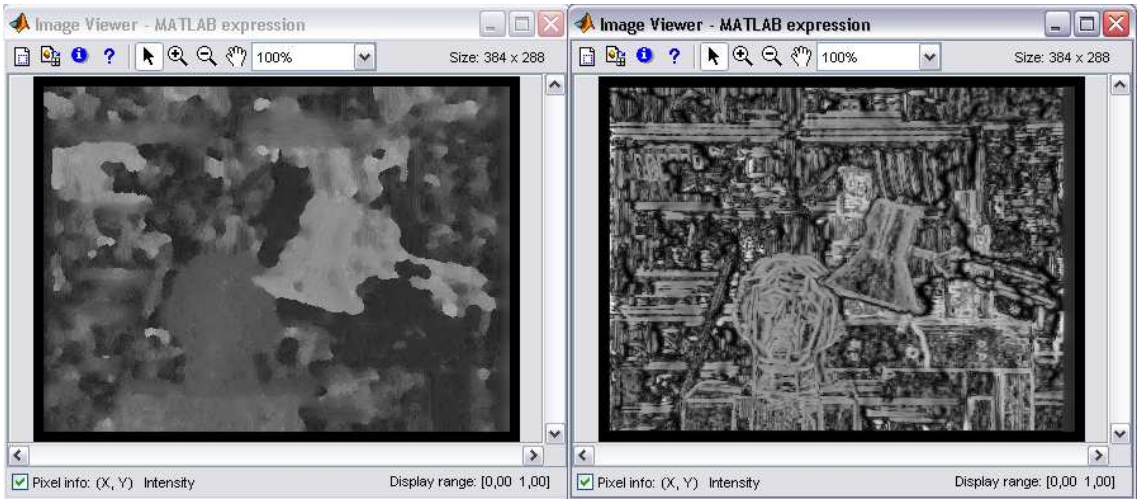

FIGURA 71 : “TSUKUBA” RADIO_LO= 1, RADIO=2, RADIO_D =1, ITERACIONES 15

Como puede verse el resultado sigue sin mejorar.

Probaremos ahora con un par estéreo que tiene una sola profundidad llamado “zizou” de 214x384.

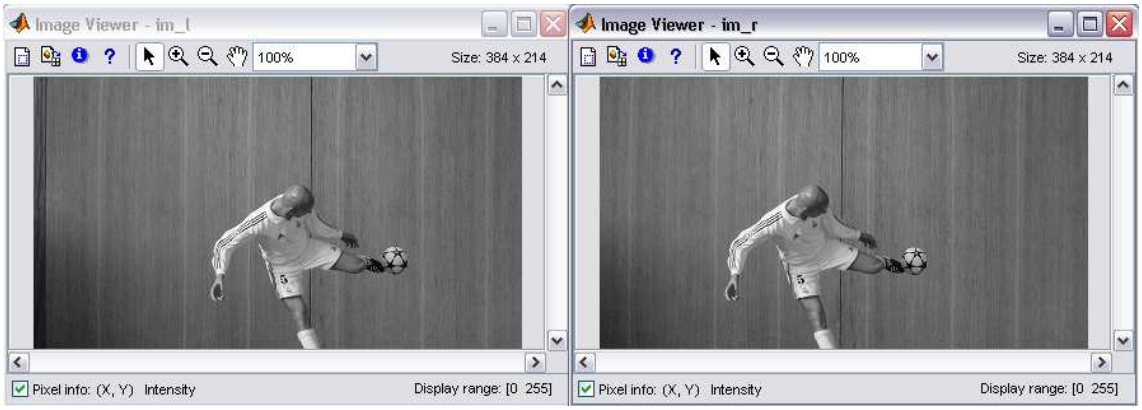

FIGURA 72 : PAR ESTÉREO “ZIZOU”

El rango de disparidades es d_min= 20 hasta d_max= 48. Con los parámetros de radio_lo=2, radio=3, radio_d=1 y 10 iteraciones.

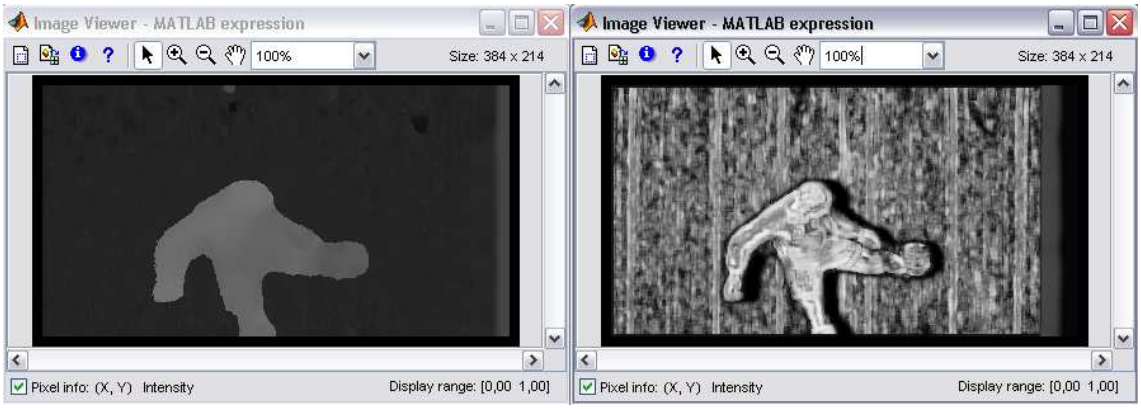

FIGURA 73 : “ZIZOU” ITERACIONES 10.

Observamos que con los parámetros escogidos los resultados son bastante satisfactorios. Vamos a probar ahora los efectos de escoger mal el rango de disparidad. Probamos ahora con d_min=25 y d_max=45.

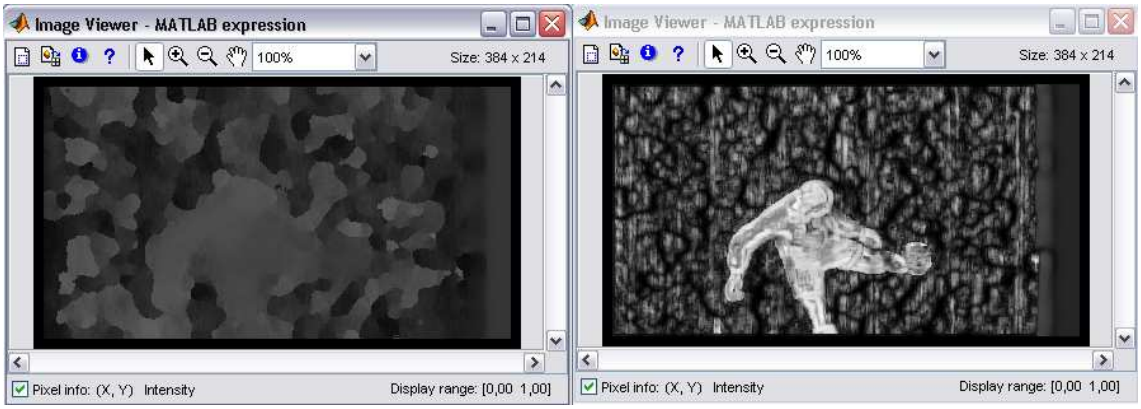

FIGURA 74 : “ZIZOU” RANGO MENOR DE DISPARIDAD

Vemos que el resultado es inaceptable, no se distingue casi ni la figura. Si observamos la imagen de la derecha podemos ver que la mayoría de los

emparejamientos del fondo están oscuros lo que quiere decir que no se han emparejado bien o que podrían ser puntos ocluidos. Esto es debido a la reducción del rango de disparidad. Comprobaremos que si ampliamos el rango más de lo debido el resultado puede empeorar, pero si lo hace no es tan negativo como si lo reducimos. Establecemos d_min=15 y d_max=55.

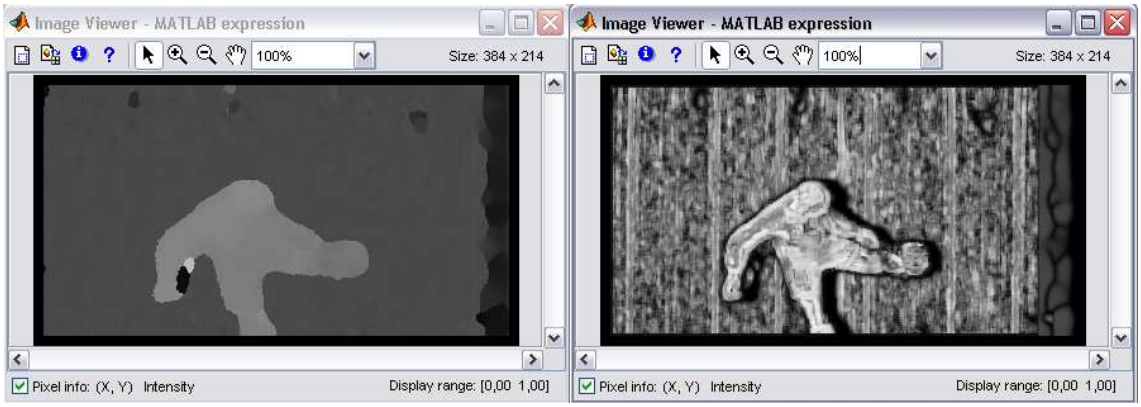

FIGURA 75 : "ZIZOU" RANGO MAYOR DE DISPARIDAD

4.3. CONCLUSIONES

Tras observar los resultados y el funcionamiento del algoritmo lo primero que cabe destacar es el elevado número de parámetros que se le deben configurar inicialmente. Esto sería un inconveniente si el propósito fuese hallar un algoritmo óptimo para hallar los mapas de profundidad de una escena dada de manera automática. Pero el fin del proyecto no es ese. Este es un paso intermedio que se puede optimizar pero que de forma general se puede decir que funciona.

Dentro de la configuración de los parámetros los más importantes a priori son el rango de disparidad sobre el que se buscaran las posibles correspondencias. Los parámetros d_min y d_max. Si se escoge un rango diferente al que corresponde la escena los resultados no son satisfactorios, como puede verse en la Figura 74. En caso de duda lo mejor es que el rango sea grande pero que abarque siempre el rango real de disparidad de la escena. Así los resultados no serán tan negativos, ver Figura 75. Para poder seleccionar el rango lo que se debe hacer es observar el par de imágenes estéreo y fijarse primero en la capa más profunda (la más alejada). Escogemos un punto que podamos luego identificar en la otra imagen y comparamos la distancia en píxeles que hay entre los dos puntos. Ese sería el valor máximo que se le debería de dar al parámetro d_min. Para establecer el valor de d_max se debería hacer lo contrario pero con un punto que esté en la capa menos profunda (la más cercana). Se compararían dos puntos homólogos que estén en la capa más cercana a la cámara, se observa su disparidad y ese sería el valor mínimo que debería tener el parámetro d_max. Ese sería el rango mínimo aceptable, si vemos que los resultados no son buenos podemos ampliarlo pero tampoco excederse.

Otro de los parámetros es el número de iteraciones. Como hemos comprobado desde la Figura 59 hasta la Figura 63 a partir de 10 (o 15 como recomiendan los autores) los resultados pueden mejorar pero no se aprecia casi la mejoría. Por eso podemos establecer este parámetro como fijo igualándolo a 10 que en nuestros resultados ha funcionado muy bien.

Los tres parámetros siguientes son los que corresponden con el radio de búsqueda, radio_lo, radio y radio_d. Los autores nos recomiendan los valores entre 1,2,1 y 2,3,1 respectivamente. Como se observa en la Figura 68 y Figura 70 los resultados son mejores ampliando el rango pero no varían demasiado. Por ello se podrían escoger inicialmente los valores de radio_lo=1, radio=2 y radio_d=1 y ejecutar el algoritmo. Si vemos que necesitamos mejores resultados procedemos a ampliar el radio a los valores máximos recomendados.

Como último parámetro configurable tenemos la posibilidad de elegir el método de búsqueda de correspondencias, mediante SAD o por correlación normalizada. Hemos visto en la Figura 61 y Figura 65 que aplicar SAD o la correlación normalizada no es demasiado determinante. Cambian un poco los resultados pero no son negativos con ningún método. El problema ha venido con la Figura 68 y Figura 69. En este par de ejemplos se puede observar como utilizar SAD empeora los resultados. Esto puede explicarse porque el par estéreo “tsukuba” tiene muchas profundidades, por ello tiene muchos puntos ocluidos y además no está demasiado texturizada. Así el método de SAD no sirve tanto para comparar los puntos como el método de correlación.

Por último debemos observar que ni los mejores resultados que hemos obtenido son óptimos. Nos alejamos bastante de los mapas de profundidad reales. Esto nos lleva a pensar que cuando separemos las imágenes por capas tendremos problemas. Tendremos capas que les hayamos asignado puntos que no le pertenecen y por consiguiente tendremos capas que le falten puntos propios. Cuando apliquemos las homografías por capas, estaremos asumiendo que la matriz H de cada capa engloba todos los desplazamientos (disparidades) de cada capa en concreto. En otras palabras, se asume que todos los puntos de una misma capa están en la misma profundidad. Por ello si aplicamos la transformación H a cada capa, tendremos puntos que no estarán bien transformados debido a que no pertenecen a una de las capas. Tendremos que ver que pasa con esos puntos y como se podría solucionar.

5. AJUSTE DE PERSPECTIVA A PARTIR DE DOS IMÁGENES DE UNA ESCENA CON PROFUNDIDAD

5.1. PRESENTACIÓN DE LOS MÉTODOS ADOPTADOS Y DE LOS POSIBLES PROBLEMAS ASOCIADOS

El objetivo de esta tercera parte del proyecto consiste en mostrar los resultados que se obtienen al realizar el proceso homográfico sobre imágenes que tienen varias profundidades.

Este proceso sobre imágenes planas (que no tienen ninguna profundidad) está bastante desarrollado y tiene una solución bien conocida y satisfactoria. Los inconvenientes que presenta el hacer homografías sobre imágenes con varias profundidades son varios. Por ejemplo el echo de tener dos imágenes estéreo de una escena con varias profundidades implica que habrá puntos en una imagen que no podamos ver en la otra y viceversa. Estos puntos como mencionamos anteriormente los llamaremos puntos ocluidos. Otro ejemplo sería también la condición de brillo de los objetos, según se refleje la luz del objeto en el objetivo de las dos cámaras se tendrán diferencias de color, contraste o apariencia. Por ello se entiende, antes de empezar a obtener resultados, que no se va ha encontrar una solución óptima.

Los pasos a seguir para poder hallar las homografías serian podemos verlos en Figura 76.

Partimos inicialmente de las dos imágenes estéreo y de la imagen de disparidad referente a la imagen de la derecha.

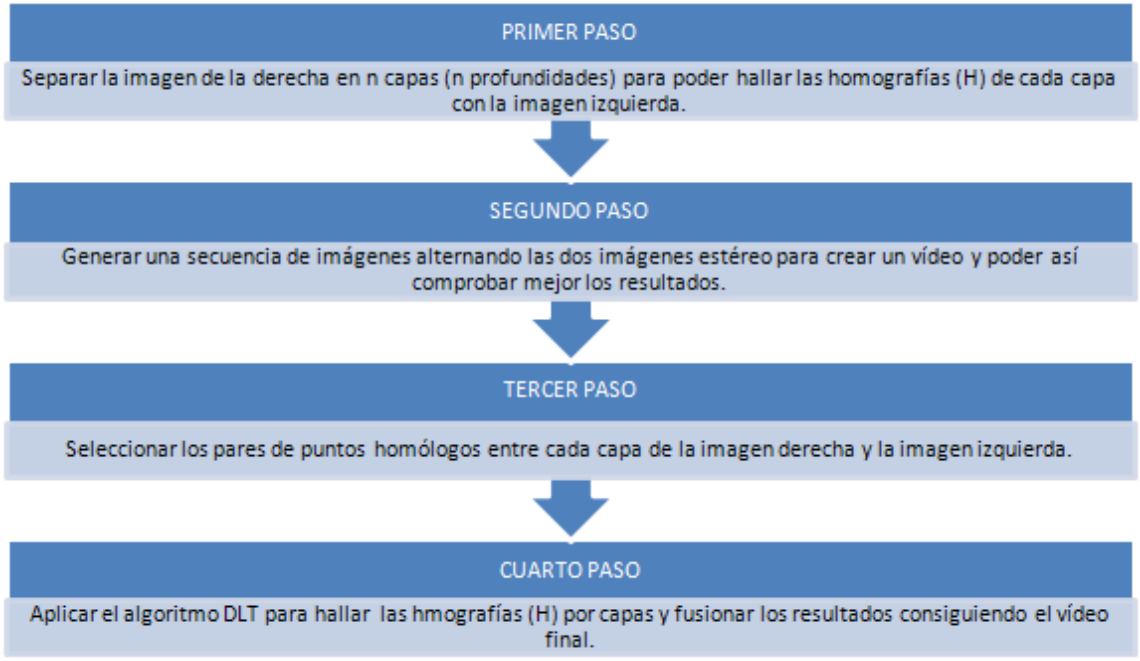

FIGURA 76 : ESQUEMA DE NUESTRO ALGORITMO

Tras la esta breve descripción empezaremos a explicar detenidamente cada paso.

5.2. IMPLEMENTACIÓN DEL ALGORITMO

5.2.1. PRIMER PASO: EXTRACCIÓN DE CAPAS: TÉCNICAS DE MEAN-SHIFT

En este primer apartado se parte inicialmente de tres imágenes: la imagen de la cámara izquierda, la imagen de la cámara derecha y el mapa de profundidad relativo a una de las dos imágenes anteriores, en nuestro caso a la imagen de la derecha.

En resumen el procedimiento seria extraer n capas de la imagen de profundidad y posteriormente obtener esas mismas capas pero de la imagen real a la que corresponda dicho mapa de profundidad.

La primera pregunta que se debe abordar es ¿como extraigo n capadas de la imagen de profundidad?

Para ello debemos fijarnos en el histograma de dicha imagen. Estas imágenes en la generalidad lo que muestran es la profundidad de la escena mediante una escala de grises. Mas menos para unos mismos puntos de una misma profundidad se les suele asociar un mismo valor (o uno cercano) de la escala de grises. Además entre una profundidad y otra también podemos observar que habría un pequeño salto entre los valores de la escala en los que encontraríamos dichas profundidades. Por ello podemos también asumir que los histogramas que extraigamos de estas imágenes van a tener una forma de picos y valles. La siguiente imagen muestra un ejemplo de lo explicado anteriormente.

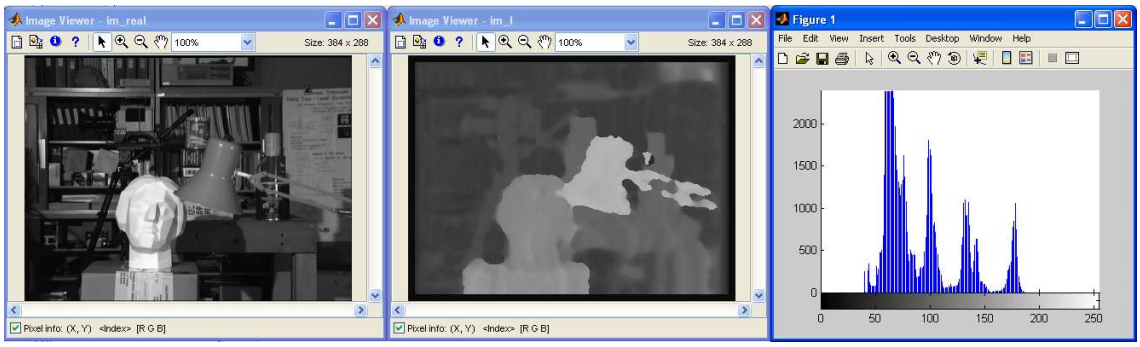

FIGURA 77 : "TSUKUBA" HISTOGRAMA DEL MAPA DE DISPARIDAD OBTENIDO EN EL CAPÍTULO DE *DISPARIDAD*.

En la imagen de la izquierda se observa la escena real. La imagen del medio muestra el mapa de disparidad (o profundidad) hallado anteriormente. La imagen de la derecha corresponde al histograma de la imagen de profundidad, se pueden observar los picos y valles descritos anteriormente.

Visto esto, lo primero que se nos ocurrió hacer fue establecer unos umbrales para separar el histograma en *n* partes, cada parte seria una de las capas. La forma de obtener las capas de la imagen de profundidad a partir de los umbrales seria trivial: los píxeles de dicha imagen que estén entre los umbrales *i* y *j* estarían en la *capa i_j* y no podrían estar en otra distinta. El resultado de la superposición de todas las capas seria la imagen de profundidad. (La nomenclatura es solo un ejemplo para entender el concepto).

La selección de estos umbrales inicialmente fue de manera uniforme. Más concretamente el procedimiento fue dividir el histograma en *n* trozos para obtener un paso y con ese paso obtener los umbrales y con ello las *n* capas de la imagen. El resultado como puede observarse en la siguiente imagen no era del todo bueno (n = #capas = 3).

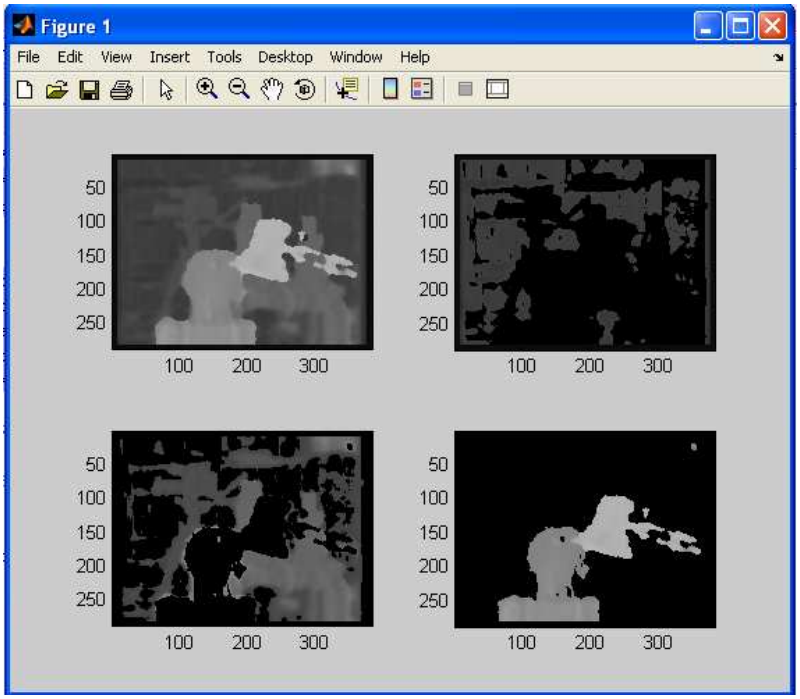

FIGURA 78 : EXTRACCIÓN DE CAPAS. PRIMER INTENTO.

El problema era que al seleccionar automáticamente un paso, este podía cortar al histograma justamente por ejemplo en un pico y así cortar una misma profundidad. Esto conllevaría tener en dos capas diferentes puntos de una misma profundidad.

Una opción sería establecer los umbrales de forma manual pero eso resulta inaceptable, así que debía escoger otro método de extracción de umbrales.

Observando el histograma el planteamiento sería, ¿en qué puntos debía establecer los umbrales si lo hiciese de forma manual? Evidentemente los mejores lugares para establecer los umbrales eran los valles, debido a que los picos representan alta concentración de puntos en un determinado rango de la escala de grises, es decir, los picos serían las capas que tenemos que separar.

A primera vista la manera mas eficiente de extraer picos y valles sería aplicando el concepto de la derivada. Si aplicamos la derivada al histograma solo tendríamos que mirar que valores son igual a cero y extraer los que correspondan a los valles.

El principal problema es que como el histograma es discreto , dentro de un mismo pico pueden existir altibajos en los valores y la derivada tendría muchos cruces por cero, y no podríamos saber cuales son picos, valles o simples altibajos.

La solución planteada es obtener la curva envolvente del histograma con un tamaño de ventana configurable. Hallada la envolvente se presenta otro problema. Debido también a que la envolvente sigue siendo discreta si aplicásemos la derivada el resultado no tendría por que tener algún valor igual a cero. Es seguro que algún cambio de positivo a negativo y viceversa tendrá pero que haya algún valor igual a cero no. Por ello en vez de aplicar la derivada se ha desarrollado un algoritmo que detecta los máximos de toda la envolvente.

El algoritmo en si es sencillo, va cogiendo las muestras de la envolvente y considera que una muestra es un máximo si la anterior y la posterior son menores que la dada.

Con ello se extraen los mayores picos, tantos como capas y se divide el historial entre los puntos intermedios entre picos.

Obtenidos ya las diferentes partes en las que queda dividido el histograma el siguiente paso sería obtener la posición de los puntos que pertenecen a cada capa. Esto es, se deben localizar en la imagen de disparidad y posteriormente en la imagen real los píxeles que pertenecen a una capa determinada. Para ello solamente es necesario comprobar entre que umbrales se encuentra el píxel en cuestión y asignárselo a la capa que corresponda.

El resultado de este proceso se puede observar en la siguiente figura:

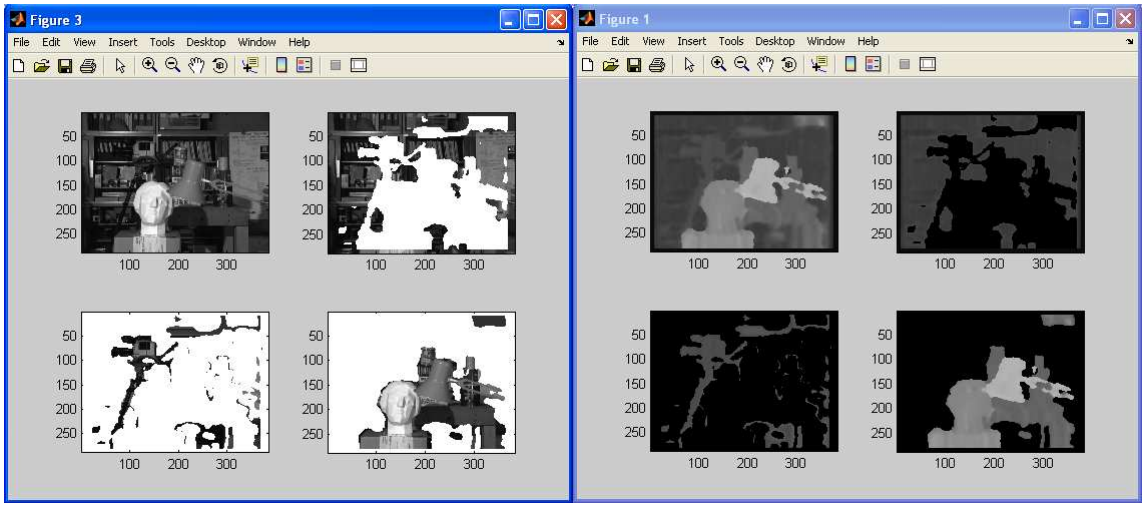

FIGURA 79 : EXTRACCIÓN DE CAPAS. SEGUNDO INTENTO.

La imagen de la derecha representa la extracción de las n capas (n = 3) Para los umbrales escogidos de manera automática. La imagen de la izquierda representa las 3 capas de la imagen real que se corresponde con la imagen de disparidad. Se aprecia que la extracción de las capas respecto al intento anterior (Figura 78) es más óptima. Se puede considerar un buen resultado.

En este punto también se obtienen unas máscaras correspondientes a las capas seleccionadas que se utilizaran posteriormente. Para el descarte de puntos que no pertenezcan a una capa determinada.

5.2.1.1. TÉCNICA DE MEAN-SHIFT

El método explicado anteriormente es de alguna forma un método casero para extraer las capas del histograma. Así que para apoyar el proyecto sobre un algoritmo contrastado decidimos probar la técnica de mean-shift.

La idea es la siguiente. Entre las muchas aplicaciones que tiene esta técnica una de ellas es la segmentación. El funcionamiento es sencillo de explicar, no se entrará en el proceso matemático asociado. La técnica de Mean-Shift establece primeramente una ventana de promediado. Esta ventana se va corriendo por el histograma (para una dimensión por ejemplo) o por la imagen (para dos dimensiones) y va obteniendo las medias centradas en cada punto. Para hacernos una mejor idea pensemos sólo en un histograma. La ventana se va corriendo de izquierda a derecha. Primeramente las medias calculadas se irán incrementando pues empezaremos por valores pequeños hasta llegar a los picos del histograma. Una vez llegue a un pico el valor medio de la ventana irá aumentado a menor ritmo hasta que se iguale de un punto a otro y a partir de ahí comience a descender su valor. Es entonces cuando al punto en el que la media está igualada se le señala como un pico. Si se sigue corriendo la ventana ocurrirá el efecto contrario. El valor de las medias calculadas dejará de descender, se igualará en algún punto y empezarán a aumentar. Ese punto donde la media se iguala se considerará un valle. Si hacemos esto por todo el histograma obtendremos de forma sencilla los picos y valles que buscábamos.

Así para una imagen general, si segmentamos su correspondiente imagen de disparidad y extraemos el valor de cada capa de la escala de grises, ya tendríamos nuestros umbrales y ya podríamos separar la imagen original en n capas. Ésta técnica hace esta segmentación de forma automática únicamente pasándole como argumentos la señal y el ancho de banda de la ventana de convergencia.

El procedimiento es sencillo. Se establece un rango creciente de anchos de banda y se va lanzando el algoritmo con cada nuevo ancho de banda (tamaño de la ventana). Así cuando el número de capas que devuelva se repita durante un número establecido de veces (nueve en nuestro caso por establecer un valor de forma empírica. Se puede configurar según creamos conveniente) se asume que ese número es el número real de capas de la imagen. De esta manera automatizamos el algoritmo ya que el único parámetro que necesita es la señal en cuestión.

Además tras la ejecución de este algoritmo obtenemos un parámetro que nos indica el clúster (o capa) a la que pertenece cada punto. Sólo tenemos que construir las máscaras a partir de este parámetro y posteriormente aplicar las máscaras a la imagen real para obtener las capas reales.

El resultado de aplicar este algoritmo sobre las mismas imágenes que estábamos probando se muestra en la Figura 80.

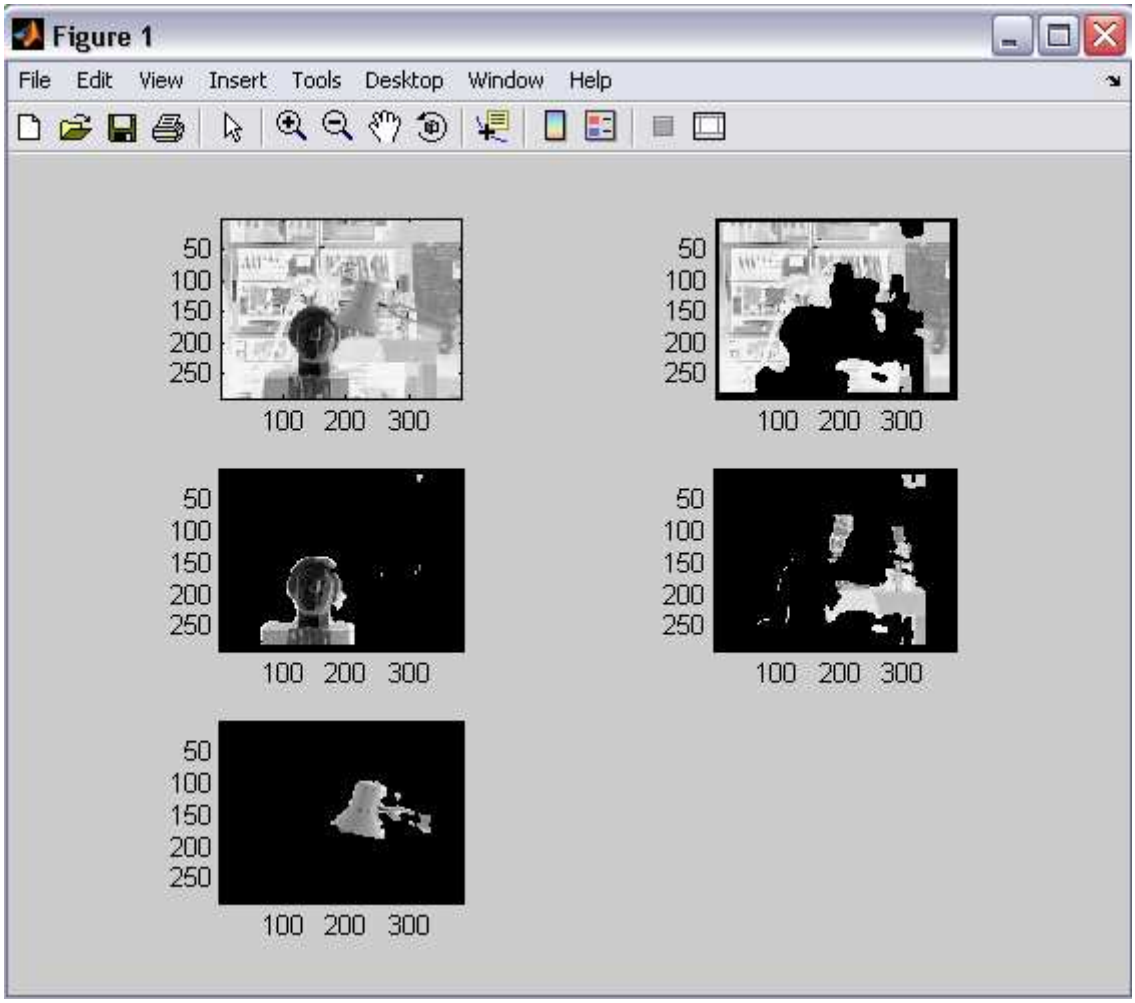

FIGURA 80 : EXTRACCIÓN DE N CAPAS MEDIANTE MEAN-SHIFT

La principal diferencia sobre nuestro método es la extracción de una cuarta capa (la que represente la mesa y los botes).

5.2.2. SEGUNDO PASO: GENERAR UNA SECUENCIA DE IMÁGENES ALTERNADAS

En este segundo apartado se parte inicialmente de 2 imágenes: la imagen de la cámara izquierda, la imagen de la cámara derecha.

Utilizando esas dos imágenes se genera una secuencia de n imágenes (con n= 50 valdría) El procedimiento es bastante sencillo y no merece la pena explicarlo.

5.2.3. TERCER PASO: SELECCIÓN DE PUNTOS HOMÓLOGOS DE FORMA AUTOMÁTICA: TÉCNICAS DE SURF

En este tercer apartado se parte inicialmente de 2 imágenes: la imagen de la cámara izquierda, la imagen de la cámara derecha. Además de las 3 capas de la imagen real correspondiente con la imagen derecha, obtenidas anteriormente en el primer paso.

El objetivo es escoger pares de puntos homólogos para cada capa y así posteriormente poder realizar la homografía por capas. Para escoger los puntos se comparan la imagen de la izquierda con cada una de las capas (las capas seleccionadas corresponden a la imagen de la derecha) y se van seleccionando los pares de puntos homólogos.

Llegados a este punto tenemos dos formas de escoger estos pares de puntos. La primera es la expuesta en la primera parte del proyecto. Es decir usar el algoritmo de Shi y Tomasi para la detección de puntos característicos y posteriormente buscarlos en la otra imagen por correlación. La segunda opción y siempre segura es usar la herramienta de matlab cpselect para seleccionar de manera manual los pares de puntos que deseemos. El inconveniente de esta segunda opción es la no automatización de los pasos. Por ello elegimos la opción desarrollada en la primera fase del proyecto.

Se deben seleccionar 9 pares de puntos homólogos por capa. En la selección de estos puntos se debe intentar que todos estén en la misma profundidad real. Esto quiere decir que como los mapas de disparidad no son exactos las capas de profundidad no son tan precisas como se desearía. Esto puede llevar a escoger dos pares de puntos homólogos en una misma capa y que no estén en la misma profundidad. En otras palabras, todos los puntos de una misma capa no tienen porque estar en la misma profundidad, de hecho lo normal es que no estén, se interpreta que más o menos si pertenecen a la misma profundidad pero no es así.

La explicación de tener que escoger 9 pares de puntos en parte es por esta diferencia de profundidades. Ya que la homografía si fuese perfecta necesitaría 4 pares de puntos pero como los pares de puntos no pertenecen todos a la misma profundidad, cuantos mas pares tengamos la homografía se ajustara mejor a la capa seleccionada. Como se comentó en el capítulo referente a la extracción de puntos homólogos consideramos el número de nueve pares de puntos correspondientes la mejor opción debido a la cuadriculación de la imagen de referencia.

El procedimiento desarrollado en la primera parte del proyecto por el cual se seleccionaba 9 pares de puntos homólogos entre dos imágenes estéreo tiene algunos inconvenientes. Se necesitaban una serie de requisitos que en este caso no se cumplen. Por ejemplo según ese procedimiento se dividía la imagen en 9 nueve cuadrantes y se extraía nueve pares de puntos, uno por cuadrante. En este caso los pares de imágenes son la imagen de la izquierda y cada una de las capas. Una capa no tiene porque tener píxeles validos en los nueve cuadrantes, por ejemplo puede tener 4 cuadrantes vacíos o con píxeles pero mínimamente distinguibles en las dos imágenes. Por ello no se puede aplicar este método para imágenes con varias

profundidades. Sin embrago la disposición del método es perfecta para imágenes planas.

Por este motivo se pensó en utilizar otro método muy utilizado en la detección de puntos característicos. La técnica se conoce como SURF y ya la describimos en el capítulo referente a la selección de puntos homólogos.

El procedimiento es el siguiente. Se aplica el algoritmo se SURF a la imagen derecha aumentando el umbral de forma que tengamos una serie no muy elevada de puntos característicos, tendríamos los más robustos. Posteriormente aplicaríamos el mismo algoritmo sobre la segunda imagen (la imagen de la izquierda) y obtendríamos una serie de puntos característicos pero sobre ésta imagen. A continuación aplicaríamos las máscaras a la imagen derecha para separarla en capas y tener cada capa con una serie de puntos característicos marcados. Gracias al descriptor podemos ir comparando parejas de puntos y ver la semejanza entre sus descriptores. Por ejemplo, tenemos los puntos de la primera capa de la imagen de la derecha y queremos ver cuales son sus correspondientes en la imagen de la izquierda. Aplicamos el algoritmo sobre la imagen derecha, extraemos la capa primera y obtenemos 9 puntos. Luego aplicamos el algoritmo sobre la imagen izquierda manejando el umbral para obtener más puntos característicos que en la imagen derecha. Posteriormente mediante un método que compare distancias, por ejemplo la distancia euclídea vamos comparando los descriptores de los puntos de la capa con los descriptores de la imagen izquierda. Y vamos estableciendo los emparejamientos según la semejanza de estos descriptores.

Tras la ejecución de este método obtendríamos nueve pares de puntos homólogos por cada capa. Además de otros nueve pares de puntos homólogos que corresponderían al emparejamiento de la imagen derecha e izquierda sin tener en cuenta las capas. Es decir como si fuesen imágenes planas, para comparar posteriormente los resultados.

5.2.4. PARTE 4: APLICACIÓN DE LAS HOMOGRAFÍAS POR CAPAS Y POSTERIOR RECONSTRUCCIÓN DE LA SECUENCIA

En este cuarto apartado se parte inicialmente de la secuencia generada en el segundo paso, de los emparejamientos de puntos homólogos hallados en el tercer paso y de las capas y mascaras halladas en el primer paso.

El funcionamiento en este paso se basaría en hallar las matrices H relativas a las parejas de imágenes formadas por cada capa y la imagen de la izquierda. Así tendríamos n matrices de trasformación, una por cada capa. Para obtener estas matrices lo único que tenemos que hacer es aplicar el algoritmo DLT detallado en el apéndice 1.

Como sabemos esta matriz H es una función entre dos imágenes que trasporta la posición de los píxeles de una imagen (la izquierda por ejemplo) a la posición que ocuparían en la otra imagen (la derecha por ejemplo). Es decir, el píxel (1,1) de la

imagen de la izquierda mediante la función H se podría trasladar al píxel (3,4) de la imagen derecha, por ejemplo. Y así con todos los píxeles de la imagen izquierda. De la misma manera y una vez aplicada la matriz H y obtenido el valor (3,4), se podría decir que el valor que va a tomar el píxel (1,1) de la imagen izquierda va a ser el (3,4) de la imagen derecha. Así podemos obtener la homografía relativa a la imagen derecha o a la imagen izquierda.

En nuestro caso hemos optado por hallar la matriz H que nos de los desplazamientos de los píxeles de la imagen de la derecha. Es decir en este caso el ejemplo podría ser:

El píxel (5,5) de la imagen de la derecha debe tener el valor que tenga el píxel de la posición (3,2) de la imagen de la izquierda (Si la matriz H fuese la misma que en el ejemplo anterior).

Si hacemos esta operación para cada píxel desde el (1,1) hasta el (h,w) (siendo h y w el alto y ancho de la imagen) obtendríamos otra matriz bidimensional de hxw en la que tendríamos los desplazamientos de los píxeles de la imagen de la derecha. En la cual el valor en la posición (5,5) seria (3,2). Con lo cual para hallar la imagen homográfica derecha lo que tenemos que hacer es ir cogiendo los valores de los píxeles de la imagen izquierda según esa matriz bidimensional a la que nombraremos como transformación proyectiva.

Uno de los problemas en este punto es que los valores de la trasformación proyectiva no suelen ser valores enteros y por lo tanto no pueden designar la posición de un píxel en concreto. Por ejemplo el valor de la transformación del píxel (5,5) no tiene porque ser (3,2), podría ser por ejemplo (3,456 , 2,198) y como es lógico no podríamos hallar el valor que debería ir en el píxel (5,5) puesto que sería el valor de la imagen de la izquierda correspondiente al píxel (3,456 , 2,198) el cual no existe. Por ello se hace uso de la función de interpolación en dos dimensiones. Así podemos aproximar el valor que tendría la imagen izquierda en el valor (3,456, 2,198) y poder dar un valor al píxel (5,5) en la imagen derecha.

Este proceso se debe realizar para las n capas. Con ello obtendríamos para cada capa su correspondiente transformación en la perspectiva de la imagen derecha. El siguiente paso sería superponer estas imágenes para formar el resultado final de la homografía. Para superponer los resultados utilizamos las máscaras de cada capa. Aplicamos cada máscara a su capa correspondiente y luego se suman todas.

Para ver el resultado con una visión mejor realizamos todos estos pasos para cada pareja de imágenes de la secuencia hallada en el apartado 2.

El resultado final son tres vídeos: el primero corresponde a la secuencia de imágenes izquierda y derecha alternadamente sin aplicarles ningún tipo de corrección, en el segundo vídeo la operación que se realiza sobre la imagen izquierda es una homografía general (como si la escena no tuviese profundidad), y en el tercer vídeo se presenta el resultado final de nuestro algoritmo.

5.3. RESULTADOS OBTENIDOS Y EXPLICACIÓN DE LOS PROBLEMAS RESULTANTES.

En éste apartado iremos mostrando los distintos resultados obtenidos e intentando explicar algunos de los problemas que se aprecian en cada resultado. Así iremos mejorando los resultados obtenidos hasta llegar a una solución bastante aceptable. Para ver el funcionamiento global, para cada par de imágenes se mostrarán la extracción de capas, la selección de puntos que se ha hecho y el resultado final. Para poder apreciar el resultado final lo mejor es ver el vídeo resultante. En éste documento por motivos obvios se mostrará el resultado únicamente presentado la imagen derecha y la transformación de la imagen izquierda.

Primeramente empezaremos con el par estéreo “cartas”, inicialmente y por coste computacional trabajamos con las imágenes reducidas para observar el resultado. Recordemos que el algoritmo de SURF tenía algunos errores en la selección de puntos para imágenes reducidas y con mucha profundidad. Podemos ver el resultado en las siguientes figuras.

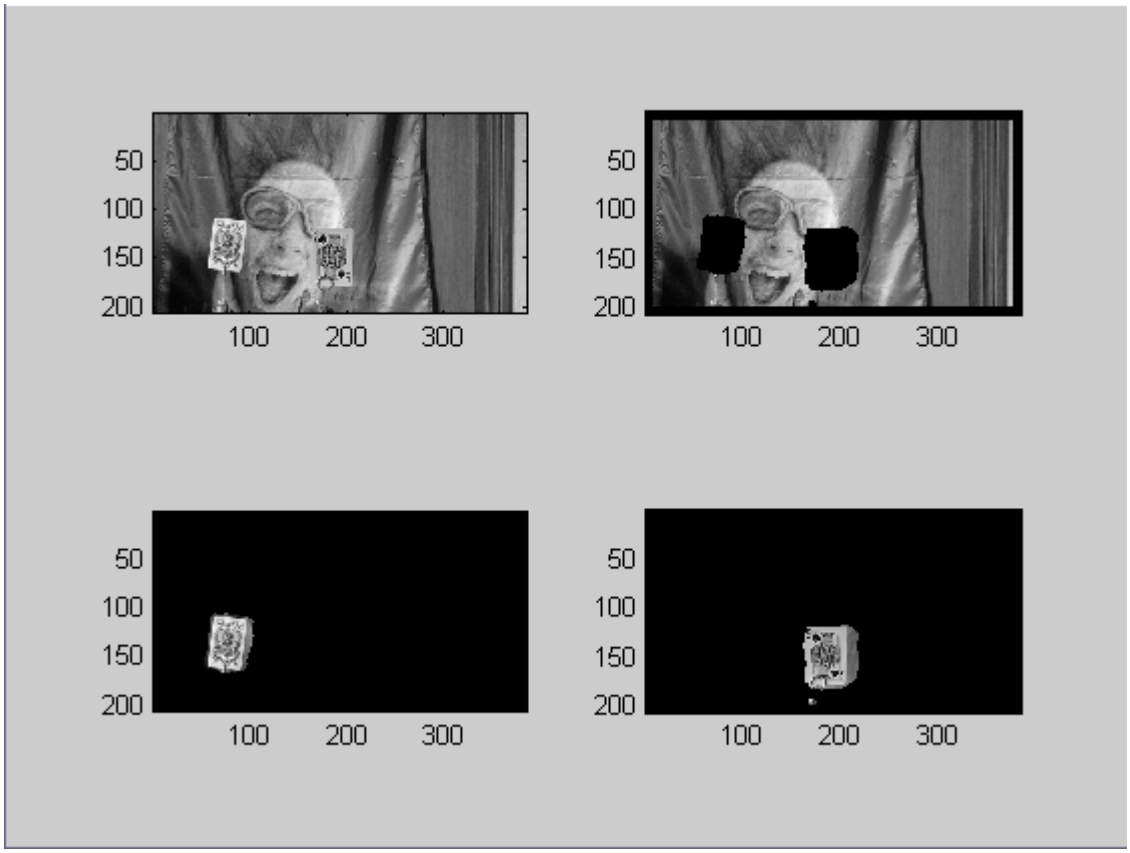

FIGURA 81 : EXTRACCIÓN DE CAPAS PARA EL PAR ESTÉREO "CARTAS" DE TAMAÑO REDUCIDO.

Para la extracción de las capas se ha usado el mapa de disparidad hallado en el apartado de disparidad referente al par “cartas” (Figura 61). La selección de puntos mediante SURF para éste par se muestra a continuación.

FIGURA 82 : "CARTAS" REDUCIDAS. SELECCIÓN DE PUNTOS MEDIANTE SURF. LA IMAGEN SUPERIOR CORRESPONDE CON LA CÁMARA IZQUIERDA Y LA INFERIOR CON LA DERECHA.

Podemos observar que la selección se ha hecho para las tres capas. Recordamos que en las capas más cercanas a cámara se producen algunos fallos en la detección de los pares de puntos homólogos. En las siguientes figuras se muestra el resultado de la fusión final. Para poder apreciar mejor los resultados se adjuntan las imágenes por separado de forma secuencial. Primero la imagen real de la cámara derecha. Segundo la imagen real de la cámara izquierda. Tercero el resultado de aplicar una homografía general a la imagen izquierda. Y cuarto el resultado final de nuestro algoritmo tras fusionar los resultados obtenidos al realizar las homografías por capas.

FIGURA 83 : "CARTAS" REDUCIDAS. IMAGEN DERECHA REAL

FIGURA 84 : "CARTAS" REDUCIDAS. IMAGEN IZQUIERDA REAL

FIGURA 85 : "CARTAS" REDUCIDAS. IMAGEN RECONSTRUIDA SIN CAPAS

FIGURA 86 : "CARTAS" REDUCIDAS. IMAGEN RECONSTRUIDA CON CAPAS

El primer problema que se observa son las réplicas de puntos que aparecen en distintas capas. Tras analizar dicho problema llegamos a la conclusión de que al aplicar una homografía distinta a cada capa, puntos de distintas capas de la imagen derecha podían requerir posiciones similares de la imagen izquierda. De ahí que se obtuvieran réplicas. Para entenderlo veamos la siguiente figura.

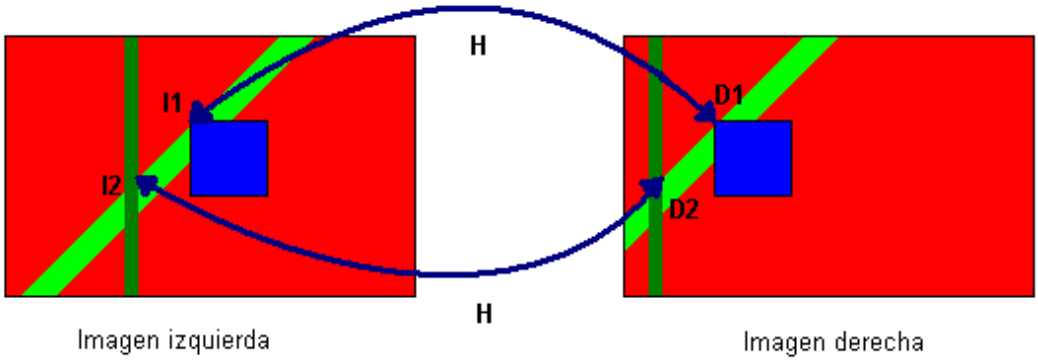

FIGURA 87 : HOMOGRAFÍA DE UN PLANO (PROBLEMA DE LOS PUNTOS REPLICADOS)

Imaginemos que las imágenes de la Figura 87 representan dos capturas estéreo de una escena que no tiene profundidad. Por ejemplo, pensemos que es un cuadro (el cuadrado azul) en una pared (la parte roja y verde). Si aplicamos el algoritmo DLT obtendríamos una homografía (H). Mediante la matriz H podríamos transformar la imagen izquierda en una imagen cuya perspectiva sería la de la cámara derecha. Así el valor del píxel que está en la posición D1 (para la imagen reconstruida) sería el valor de I1 de la imagen izquierda. Lo mismo ocurriría para el píxel I2 y su correspondiente valor en la posición de D2. Puesto que la escena no tiene profundidad para cada posición de la imagen reconstruida (que se corresponde con la perspectiva de la cámara derecha) tendremos un valor único en la imagen izquierda. Ésta afirmación no es rigurosamente cierta puesto que la homografía (H), como vimos anteriormente, generalmente trabaja a nivel subpíxel en la imagen izquierda. La reconstrucción la realizábamos por interpolación. Pero para entender el porqué del error que estamos viendo nos vale esa afirmación.

Ahora veremos que pasa si tenemos una escena con profundidad como la que podemos imaginar en la siguiente figura.

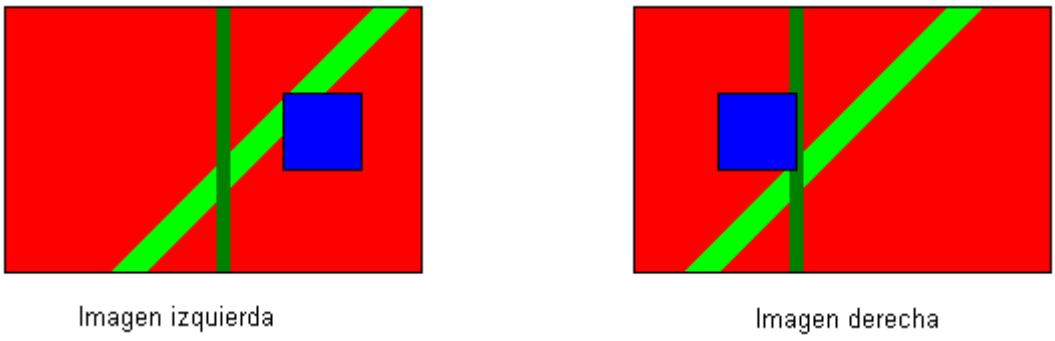

FIGURA 88 : EJEMPLO DE PAR ESTÉREO CON PROFUNDIDAD (PROBLEMA DE LOS PUNTOS REPLICADOS)

Imaginemos éste par de imágenes como un par de imágenes estéreo reales con profundidad. El fondo de la escena sería la parte roja y verde, y el objeto que está más cerca de la cámara correspondería al cuadrado azul. Como puede observarse el cuadrado azul oculta puntos en las dos imágenes, y la disparidad entre los cuadrados

es mayor (están a menor profundidad respecto a la cámara) que la disparidad entre los puntos de la capa del fondo. Visto el ejemplo veamos como serían los emparejamientos entre los puntos de la capa más cercana, los del cuadrado azul.

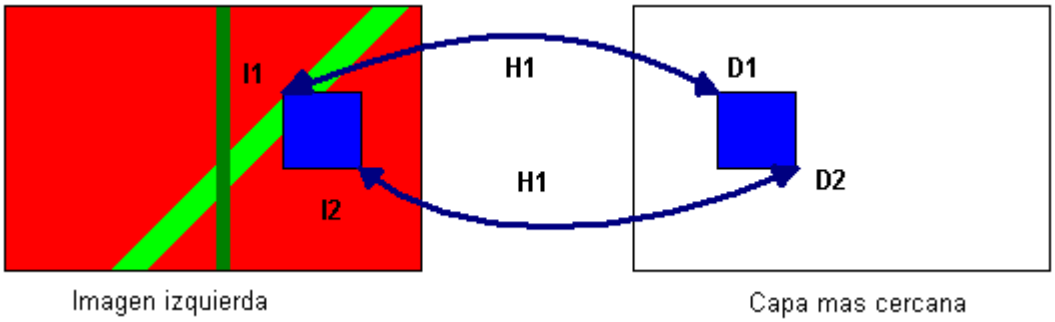

FIGURA 89 : EMPAREJAMIENTOS DE LA CAPA MÁS CERCANA (PROBLEMA DE LOS PUNTOS REPLICADOS)

Podemos ver como los puntos reconstruidos en las posiciones de D1 y D2 serán mediante la homografía H1 el valor de los puntos I1e I2 de la imagen izquierda (respectivamente). El problema viene con la capa del fondo. Veamos sus emparejamientos.

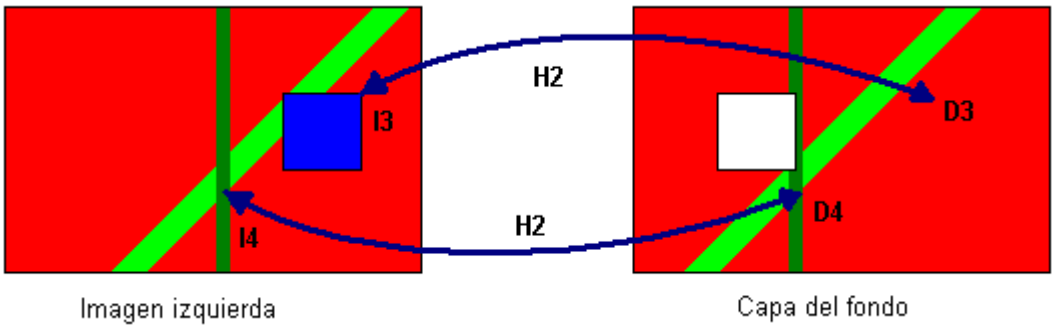

FIGURA 90 : EMPAREJAMIENTOS DE LA CAPA MÁS ALEJADA (PROBLEMA DE LOS PUNTOS REPLICADOS)

En la figura anterior podemos comprobar el motivo de que se produzcan puntos replicados. Debido a que cada capa tiene su propia homografía, H1 para la capa más cercana y H2 para la capa del fondo, hay puntos de la imagen izquierda que son transformados en más de uno para la imagen derecha. El ejemplo de éste comentario se puede observar el la Figura 89 y la Figura 90. El cuadrado azul se reconstruye perfectamente en la Figura 89 pero es replicado incorrectamente en la Figura 90. Así el resultado de reconstruir la capa del fondo sería el siguiente.

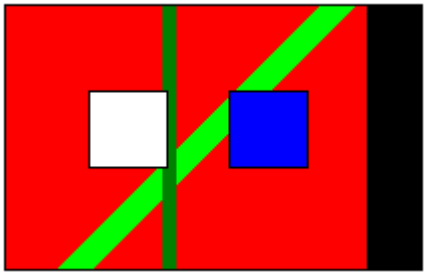

FIGURA 91 : RESULTADO DE LA RECONSTRUCCIÓN PARA LA CAPA MÁS PROFUNDA (PROBLEMA DE LOS PUNTOS REPLICADOS)

La parte negra de la derecha corresponde con puntos ocluidos. En nuestras pruebas ponemos un recuadro negro por ésta razón. Como podemos observar el cuadrado azul se replicaría entero en la capa del fondo. Si fusionásemos las dos homografías tendríamos como un resultado final la figura siguiente.

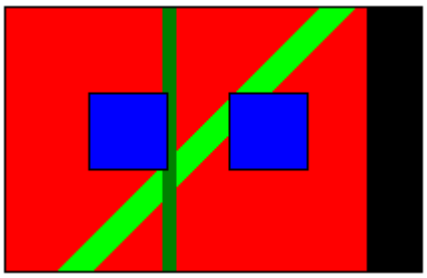

FIGURA 92 : RESULTADO DE LA RECONSTRUCCIÓN TOTAL (PROBLEMA DE LOS PUNTOS REPLICADOS)

Con éste resultado se puede comprender el problema de los puntos replicados que nos aparecían en las pruebas anteriores con el par de imágenes “cartas”. Visto el problema nos planteamos como podemos solucionarlo. La decisión fue crear una matriz del mismo tamaño que la imagen de la izquierda e ir marcando los puntos que se referencian, así cuando un punto ya se ha referenciado no se puede referenciar más. Es decir, siguiendo el ejemplo anterior de las imágenes sintéticas con profundidad, primero aplicaríamos la homografía H1 para obtener la reconstrucción del cuadrado azul. Las posiciones de los puntos que pertenecen al cuadro azul en la imagen izquierda son marcadas en la matriz que hemos creado. Luego al aplicar la homografía H2 sobre la capa del fondo, cuando se busque un valor en la imagen izquierda éste punto estará marcado y no se cogerá. La cuestión es qué hacer entonces con los puntos de la capa derecha del fondo que deberían contener el cuadrado azul. Nuestra solución fue replicar los puntos reales de la imagen derecha. Somos conscientes de que estos puntos son falsos pero gracias a la matriz de los puntos marcados podríamos ver de inmediato qué puntos estamos considerando como falsos. Asociado a ésta metodología tenemos un problema menor. Resulta a la hora de marcar los puntos en la matriz de “*puntos repetidos*”.

Como explicamos en éste capítulo la matriz H que transforma los puntos de una imagen a otra generalmente da como resultados valores decimales (trabaja a nivel subpíxel), es decir, la operación no transforma (por ejemplo) el píxel (1,1) en el píxel (4,3). La transformación sería, por ejemplo, del píxel (1,1) al píxel (4.12, 3.85). Por ello

al tener que marcar los puntos en la matriz de *puntos repetidos* debemos de redondear éstos valores. El problema de esto es que algún píxel se puede quedar sin marcar porque al redondear simplemente no se escoja, esto haría que una posterior homografía con otra H distinta, llegase a ese píxel y al no estar marcado lo referenciara como suyo cuando en realidad no lo es. Por ello para marcar los puntos lo que se debe hacer es marcar todos los píxeles del contorno. Para el ejemplo anterior se marcarían los píxeles (4,3), (4,4), (5,3) y (5,4). Y como último paso, entre los puntos de una misma capa no se deben marcar los puntos porque si no al buscar píxeles próximos habría puntos que encontrarían posiciones marcados y no obtendrían ningún valor en la imagen reconstruida.

Después de aplicar ésta solución sobre el par de imágenes “cartas” reducidas obtenemos el siguiente resultado.

FIGURA 93 : "CARTAS" REDUCIDA. EXTRACCIÓN DE PUNTOS REPETIDOS

Se comprueba que en comparación con el resultado de la Figura 86 ahora no aparecen puntos replicados. Podemos ver a continuación la matriz de puntos repetidos hallada.

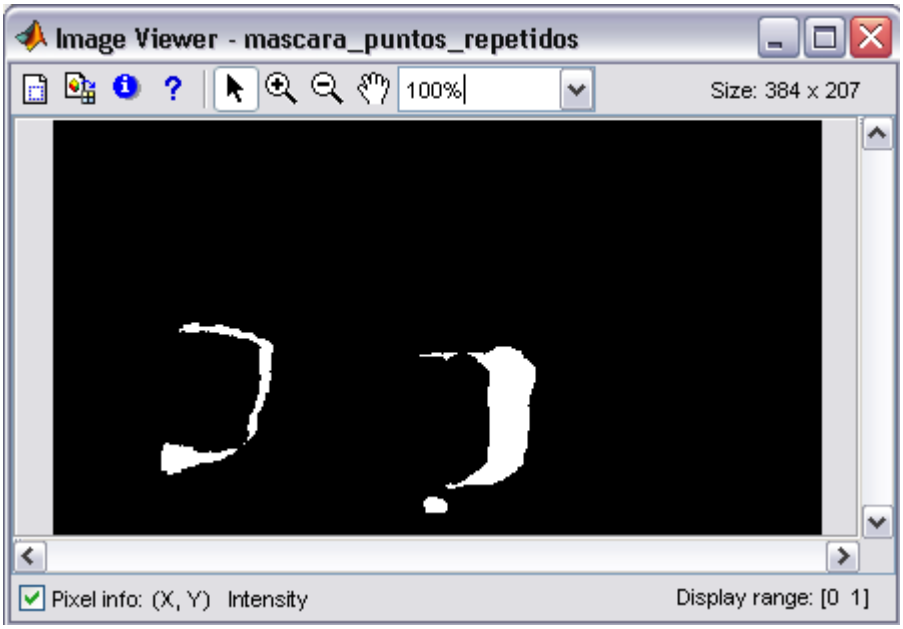

FIGURA 94 : "CARTAS" REDUCIDAS. MATRIZ DE PUNTOS REPETIDOS

Los puntos que aparecen en blanco corresponden con los puntos repetidos. Todos ellos serán los puntos falsos que tendremos en la imagen reconstruida. De hecho si nos fijamos, los puntos replicados son aquellos que están presentes en la imagen derecha pero están ocluidos en la imagen izquierda.

Cierto es que el resultado es mucho mejor, pero se siguen observando fallos nada tranquilizantes. El siguiente problema que abordamos como se observa en la Figura 93 se debe al contorno que rodea el par de cartas. Éste contorno está relacionado con la fase en la que generamos el mapa de disparidad. Puesto que el mapa de disparidad no es perfecto, hay puntos que se asocian a capas en las que realmente no están. Éste problema ya se comentó anteriormente. Veamos por ejemplo la capa de la “carta de póker”.

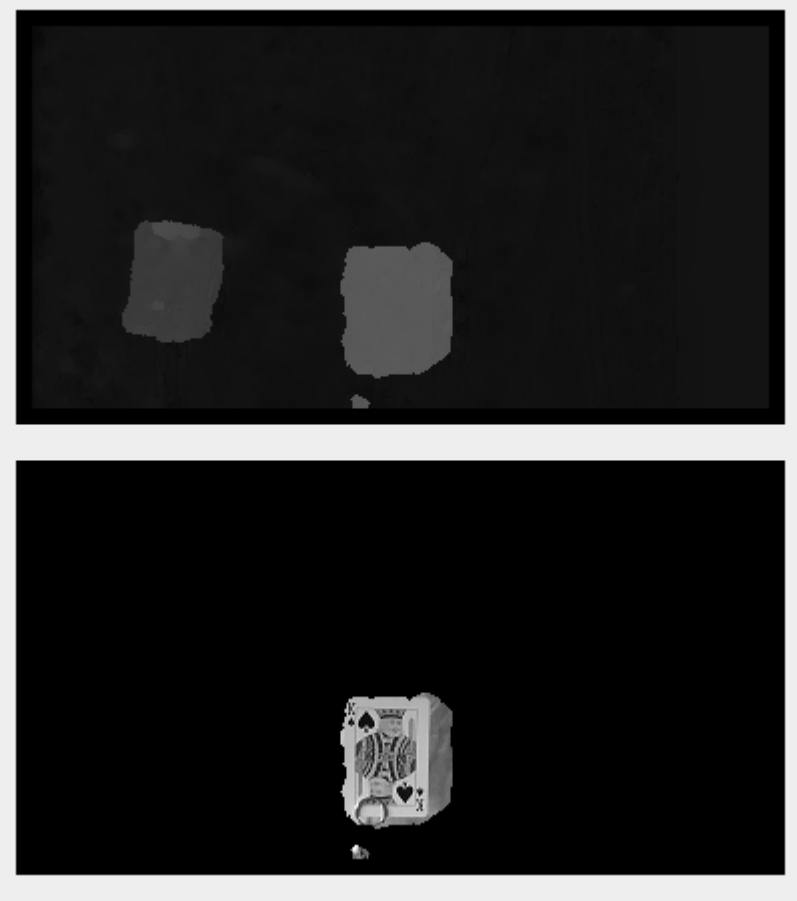

FIGURA 95 : "CARTAS" REDUCIDAS. EXTRACCIÓN DE LA CAPA MÁS CERCANA

Podemos observar que al no ser perfecta la generación de los mapas de profundidad, tenemos puntos en ésta capa que en realidad pertenecen al fondo. Debido a esto la reconstrucción de éstos puntos se realizará mediante la homografía que afecte a la capa de la “carta de póker” en vez de reconstruirlos mediante la homografía que transforma los puntos de la capa del fondo. La solución a éste problema es fácil de plantear pero casi imposible de resolver en la actualidad. Sería obtener un mapa de profundidades perfecto. Por ello hemos hallado el mapa de profundidades real de la escena, referente a la imagen derecha. La forma de hallarlo a sido “a mano” con la finalidad de ver el resultado final en el caso de que algún día pudiésemos obtener un mapa de profundidad perfecto de forma automática. En la siguiente figura se muestra éste mapa de profundidad así como cada una de las capas extraídas.

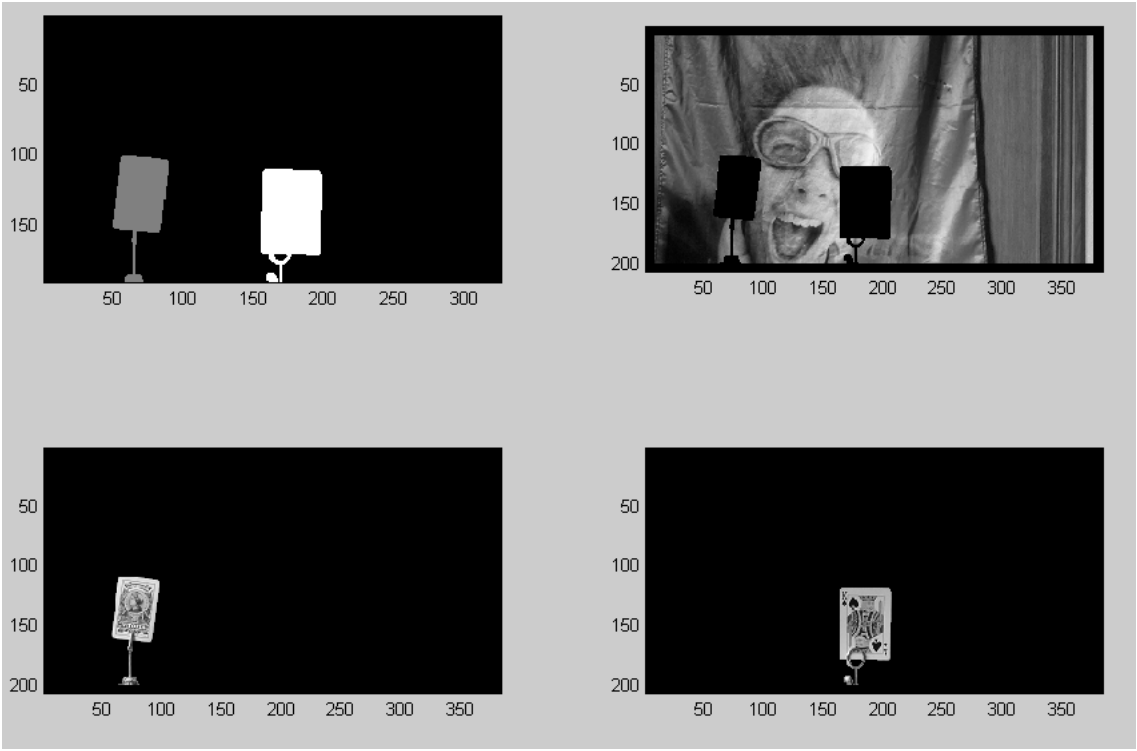

FIGURA 96 : "CARTAS" REDUCIDAS. EXTRACCIÓN DE CAPAS CON MAPA DE PROFUNDIDAD REAL.

Podemos ver que la extracción de capas ha sido perfecta. El siguiente paso sería extraer las parejas de puntos homólogos por cada capa. Mostramos el resultado a continuación.

FIGURA 97 : "CARTAS" REDUCIDAS. EXTRACCIÓN DE PUNTOS DESPUÉS DE EXTRAER LAS CAPAS CON EL MAPA DE PROFUNDIDAD REAL. (CÁMARA IZQUIERDA ARRIBA Y CÁMARA DERECHA ABAJO).

Podemos observar que la selección de puntos es la misma que antes. La extracción de las capas con el mapa de disparidad real no afecta a la selección de puntos. Esto es porque la selección de puntos se realiza sobre toda la imagen y posteriormente se le aplica la máscara de cada capa para quedarnos solo con los puntos de la capa con la que estemos trabajando. Procedemos a realizar la fusión de los resultados que obtenemos después de realizar las homografías por capas.

FIGURA 98 : "CARTAS" REDUCIDAS. RESULTADO DE LA FUSIÓN CON EL MAPA DE DISPARIDAD REAL. (CÁMARA IZQUIERDA ARRIBA Y CÁMARA DERECHA ABAJO)

Podemos observar que el problema de los bordes se ha solucionado bastante bien. Pero aparece otro problema asociado. Parece como si se siguieran produciendo puntos replicados. En realidad no son puntos replicados. Sabemos que la técnica de hallar la homografía H se basa en optimizar un conjunto de ecuaciones en las que están implicados los pares de puntos que se le pasen como argumento. Con optimizar nos referimos a hallar la solución que mejor se ajuste a ese conjunto de puntos homólogos que hemos hallado. Por ello la homografía no será perfecta y los errores pueden verse reflejados en los bordes de los objetos. Esto es porque si la homografía fuese perfecta, todos los puntos de la “carta de póker” (por ejemplo) de la imagen izquierda, tras la transformación, coincidirían con todos los puntos de la “carta de póker” de la imagen derecha. Al no ser perfecta, ésta trasformación se puede desplazar un poco en cualquier dirección (arriba, abajo, izquierda o derecha). Por eso uno o dos de los márgenes del objeto no serán referenciados por la transformación correspondiente. Se puede ver claramente en la “carta del As de oros”. Los márgenes derecho e inferior no han sido referenciados por la homografía que afecta a la carta en cuestión. Se referencian posteriormente con la homografía de la capa del fondo. Comentamos también el otro problema que llevamos viendo desde el principio y que creemos que puede tener la misma solución que éste último. El fallo en cuestión es el

que se presenta en la “carta del as de oros”. Se observa como si la carta estuviese deformada. En la “carta de póker”, aunque en menor medida también se observa éste resultado. El problema viene de trabajar con imágenes de menor resolución. Vimos que la selección de puntos homólogos no era óptima, por ello el cálculo de las correspondientes homografías (H) sería erróneo.

Una de las posibles soluciones a estos dos problemas sería trabajar con imágenes de mayor resolución. Si trabajamos con mayor resolución la selección de puntos sabemos que es más correcta, solventaríamos en parte el segundo error. Además ésta selección de puntos es mucho mas precisa y obtendríamos unas homografías mas perfectas. Con lo cual el primer problema podría quedar solucionado también. Probaremos entonces con la mayor resolución del par estéreo “cartas”. Además utilizamos el mapa de profundidad real que hemos creado “a mano”.

La selección de capas es la siguiente.

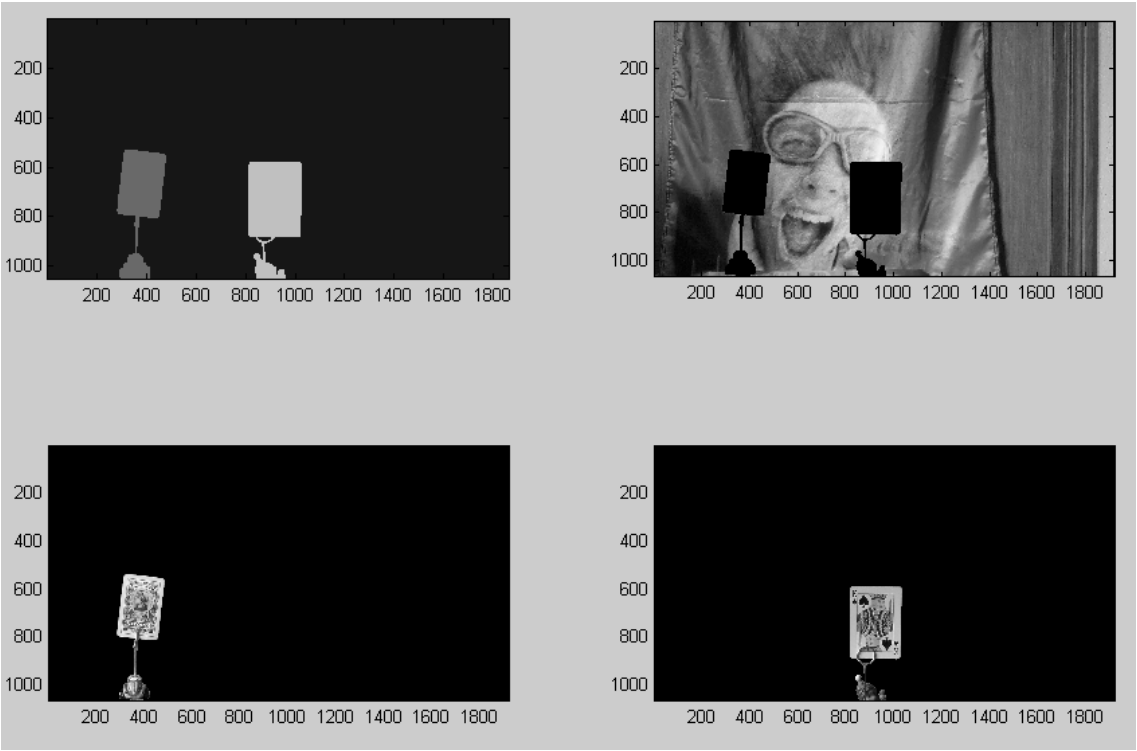

FIGURA 99 : "CARTAS" MÁXIMA RESOLUCIÓN. EXTRACCIÓN DE CAPAS CON MAPA DE PROFUNDIDAD REAL.

Podemos comprobar que la extracción de capas es perfecta. Veamos ahora la selección de puntos homólogos.

FIGURA 100 : "CARTAS" REDUCIDAS. EXTRACCIÓN DE PUNTOS DESPUÉS DE EXTRAER LAS CAPAS CON EL MAPA DE PROFUNDIDAD REAL. (CÁMARA IZQUIERDA ARRIBA Y CÁMARA DERECHA ABAJO).

Se puede comprobar que la selección de puntos homólogos es bastante satisfactoria. Probemos a realizar el proceso de fusión para obtener el resultado final.

FIGURA 101 : "CARTAS" MÁXIMA RESOLUCIÓN. RESULTADO DE LA FUSIÓN CON EL MAPA DE DISPARIDAD REAL.

Mostramos también la máscara de puntos repetidos.

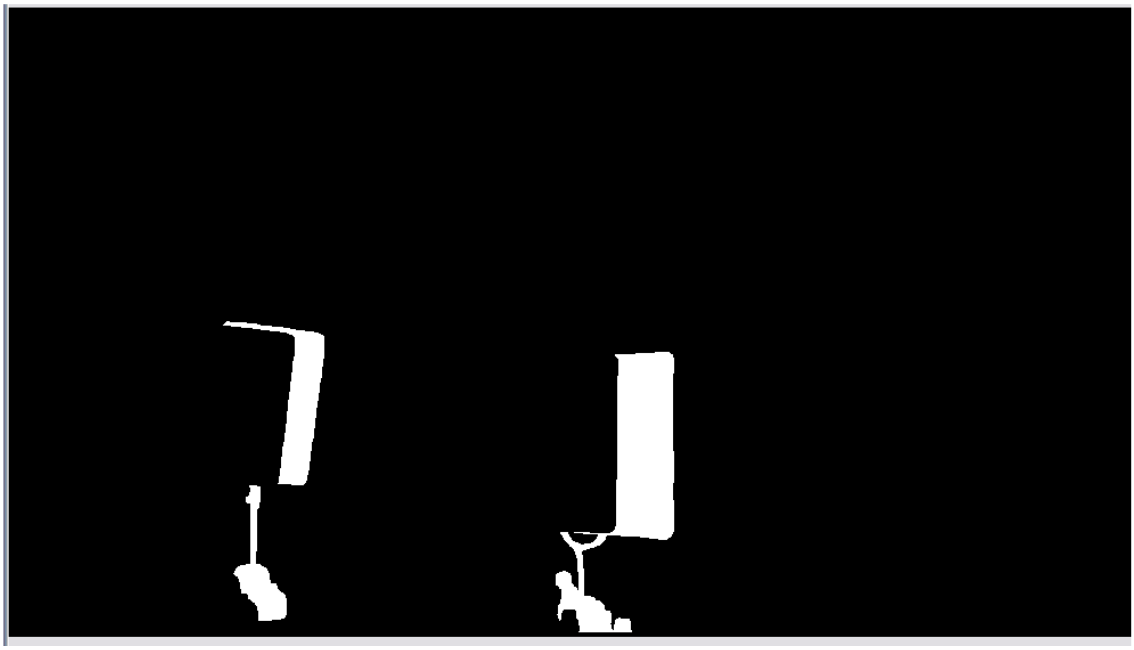

FIGURA 102 : "CARTAS" MÁXIMA RESOLUCIÓN. MÁSCARA DE PUNTOS REPETIDOS UTILIZANDO EL MAPA DE PROFUNDIDAD REAL.

Observando la Figura 101 podemos apreciar que el resultado es bastante aceptable después de ir solventando los errores que hemos comentado. Pero se siguen observando problemas con los bordes de las “cartas”. Como mencionamos anteriormente éstos problemas se deban a la pequeña inexactitud de las homografías halladas. Así pues el problema a solventar sería como mejorar la técnica para hallar una homografía perfecta. El algoritmo DLT consigue éstas homografías de forma óptima si las cámaras con las que se toman las imágenes están bien calibradas y además las imágenes tomadas están rectificadas. Éste par estéreo llamado “cartas” está tomado por una misma cámara convencional, desplazándola espacialmente. Por ello la selección de puntos homólogos puede tener algún error. Aunque visualmente ésta selección resulte óptima pueden variar en uno o dos píxeles, por ejemplo, y con ello cambiar la homografía (H) calculada. Éste hecho es el que produce que en los bordes de las capas podamos tener problemas. Por ello, para comprobar el buen funcionamiento de nuestro algoritmo, hemos creado un par de imágenes estéreo, sintéticas y rectificadas. El resultado esperado debería ser que no se obtuvieran los problemas con los bordes de las capas descritos anteriormente.

El par de imágenes sintéticas es el siguiente.

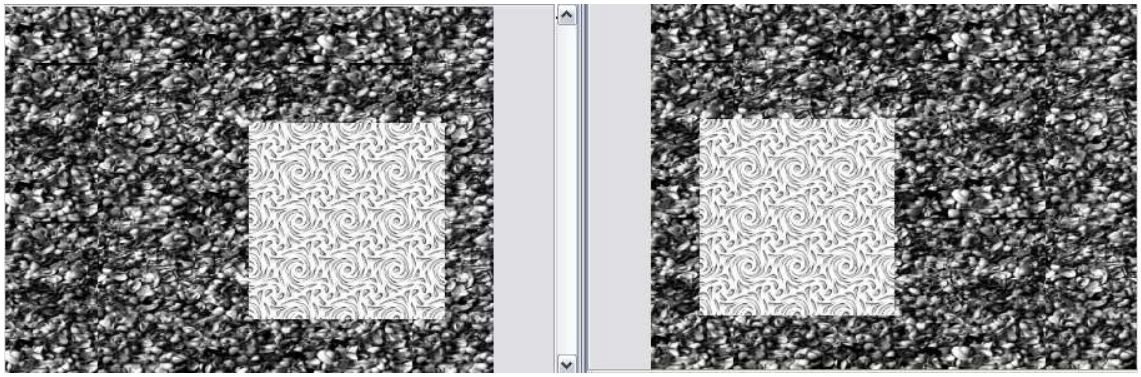

FIGURA 103 : IMÁGENES SINTÉTICAS. CÁMARA IZQUIERDA (IMAGEN IZQUIERDA) Y CÁMARA DERECHA (IMAGEN DERECHA)

A continuación se muestra el mapa de profundidad real, correspondiente a la imagen derecha.

FIGURA 104 : IMÁGENES SINTÉTICAS. MAPA DE PROFUNDIDAD REAL.

Mostramos el resultado final de la fusión de todos los resultados.

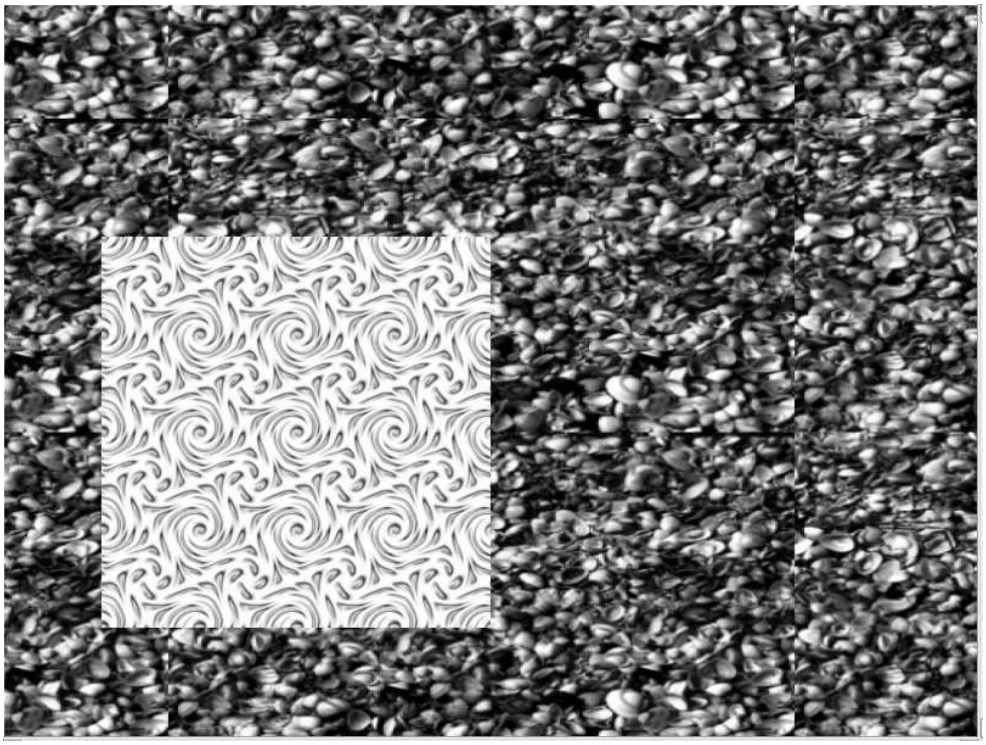

FIGURA 105 : IMÁGENES SINTÉTICAS. IMAGEN DERECHA REAL.

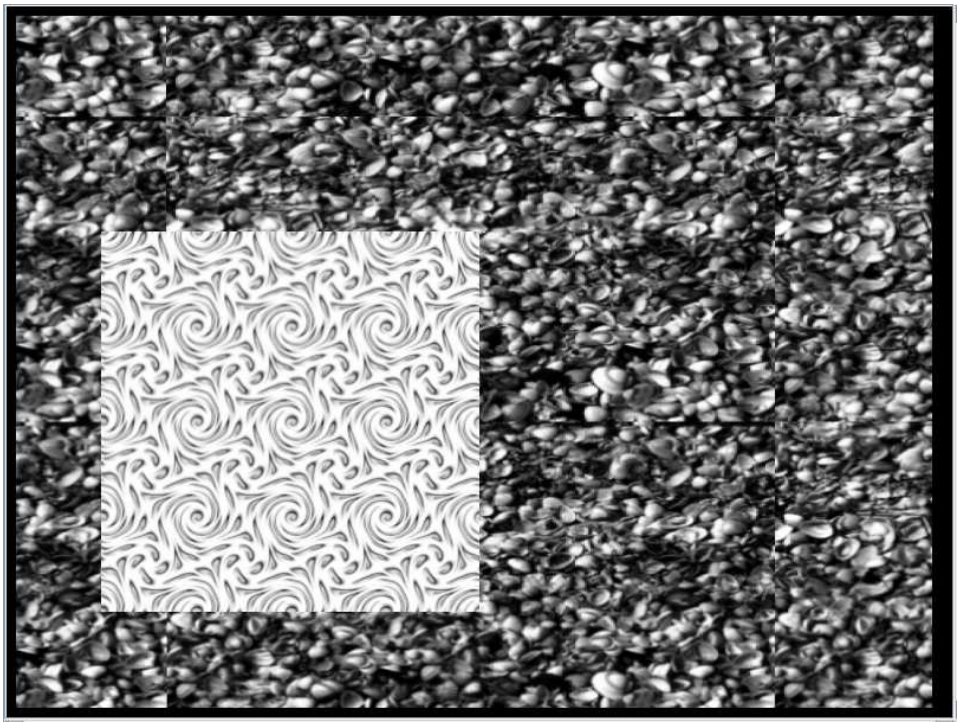

FIGURA 106 : IMÁGENES SINTÉTICAS. RESULTADO FINAL DE LA RECONSTRUCCIÓN.

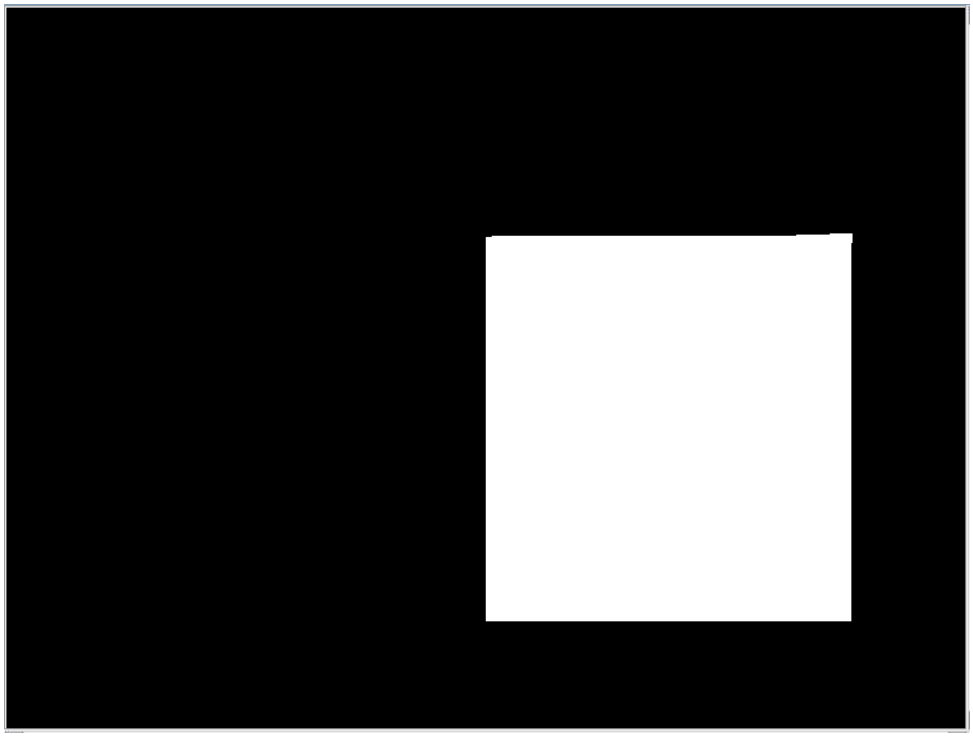

FIGURA 107 : IMÁGENES SINTÉTICAS. MATRIZ DE PUNTOS REPETIDOS.

Como se puede comprobar el resultado es bastante satisfactorio. La imagen reconstruida se asemeja mucho a la imagen real derecha. Si observamos el video creado con la secuencia podemos apreciar una ligera diferencia. Esto se debe a la fase de interpolación que utilizamos para determinar la reconstrucción de cada capa.

Al usar el método de interpolación el píxel reconstruido se parece mucho al píxel real pero no son iguales. Como comentamos, éste efecto se puede apreciar en la secuencia de vídeo. Comparando los resultados en las imágenes anteriores casi no se puede apreciar ésta diferencia. Hemos de tener en cuenta de que en la imagen reconstruida tenemos píxeles “inventados” o falsos, éstos corresponden con el cuadrado blanco de la imagen que representa la matriz de puntos repetidos (Figura 107).

6. CONCLUSIONES FINALES Y TRABAJO FUTURO

6.1. CONCLUSIONES

Después de realizar todas las pruebas descritas en los capítulos anteriores, podemos concluir que el objetivo final del proyecto se ha cumplido. La reconstrucción de la escena de profundidad ha sido llevada a cabo de forma satisfactoria. Para ello tenemos que destacar algunas condiciones que son imprescindibles para llegar al resultado final:

Primeramente debemos asumir que la reconstrucción de la escena no es cien por cien óptima. Tendremos puntos en la reconstrucción que serán puntos falsos. Analizando estos puntos falsos podemos comprobar que son precisamente los puntos de la imagen derecha que están ocluidos en la imagen izquierda. La buena noticia es que podemos saber inmediatamente que puntos son éstos mediante la matriz de puntos repetidos.

El segundo punto a tener en cuenta tiene que ver con los mapas de disparidad. Hemos comprobado que para obtener la fusión total de los resultados obtenidos debemos de trabajar con un mapa de profundidad perfecto. El problema es que actualmente conseguir ese mapa es imposible. Por ello debemos de seguir asumiendo que si automatizamos todos los pasos, tendremos errores en la imagen reconstruida debido a la no perfección del método de disparidad.

Como tercer aspecto que tenemos que resaltar se produce en la selección de puntos homólogos. Si ésta selección de puntos no es óptima la homografía (H) calculada nos proporcionará errores en los puntos reconstruidos. Por ello debemos asegurarnos que la captura de las imágenes sea óptima, además de rectificar posteriormente las imágenes. Debemos considerar que aunque la homografía (H) sea perfecta debido al echo de utilizar la función de interpolación tendremos pequeños errores que solo se podrán apreciar en la secuencia de vídeo. Éste hecho es insalvable debido a que una imagen por mucha resolución que tenga siempre será discreta, debemos utilizar la técnica de interpolación para determinar el valor a nivel subpíxel.

6.2. TRABAJO FUTURO

Se propone mejorar los principales métodos que hemos desarrollado. En particular se propone mejorar la fase de detección de puntos homólogos de forma que se puedan seleccionar de manera automática para todo tipo de imágenes, sin importar el tamaño, las formas o las texturas encontradas.

También sería muy conveniente indagar en el amplio tema de la disparidad. Hallar un mapa de profundidades real es algo que actualmente es imposible pero esa sería nuestra meta.

Como trabajo final se propone mejorar el coste computacional total, para poder utilizar éste método en tiempo real.

7. BIBLIOGRAFÍA

[1] Richard Hartley y Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[2] Yao Wang, Jörn Osteermann y Ya-Qin Zhang. *Video Processing and comunications*. Prentice-Hall, 2001.

[3] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, 1993.

[4] Olivier Faugeras y Quang-Tuan Luong. *The geometry of multiples images*. MIT Press, 2001.

[5] http://iie.fing.edu.uy/investigacion/grupos/gti/cursos/egvc/

[6] Javier Enebral González "*Detección y asociación automática de puntos característicos para diferentes aplicaciones*", Trabajo de fin de carrera, 2009.

[7] Moravec H., "Towards Automatic Visual Obstacle Avoidance", *Proc. 5th International Joint Conference on Artificial Intelligence*, pp. 584 (1977).

[8] Harris C. and Stephens M., "A combined corner and edge detector", *Plessey Research Roke Manor*, 147-151 (1988).

[9] Jianbo Shi and Carlo Tomasi., "Good Features to Track", *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593 - 600.

[10] Canny J., "A Computational Approach To Edge Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence,* 8:679-714 (1986).

[11] Lindeberg T., "Scale-Space Theory in Computer Vision", *Kluwer Academic Publishers,* (1994).

[12] Lowe D., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision* (2004).

[13] Vedaldi A., "An implementation of SIFT detector and descriptor", *UCLA CSD Tech. Report 070012* (2006).

[14] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in In ECCV, pp. 404–417, 2006.

[15] Lecumberry F., "Cálculo de disparidad en imágenes estéreo, una comparación", *III Workshop de Computación Gráfica, Imágenes y Visualización* (2005).

[16] D. Marr y T. Poggio. A *computational theory of human stereo vision*. Proc R Soc Lond, pp.301-328. May 1979

[17] José M. Lopez / Antonio Fernández Caballero / Miguel A.Fernández "*Conceptos y técnicas de estereovisión por computador*" en Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Número 027.

[18] Yuichi Ohta y Takeo Kanade. *Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Marzo 1985

[19] Yuri Boykov y Vladimir Kolmogorov*. An Experimental Comparison of Min-cut/Max-ow Algorithms for Energy Minimization in Vision.* Energy Minimization Methods in Computer Vision and Pattern Recognition, Third International Workshop, 2001

[20] Yuri Boykov y Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence,2004

[21] Vladimir Kolmogorov y Ramin Zabih. Computing Visual Correspondence with Occlusions via Graph Cuts. International Conference on Computer Vision,2001

[22] Sebastien Roy y Ingemar J. Cox. A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem. International Conference on Computer Vision, 1998

[23] Yuri Boykov, Olga Veksler y Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence,

[24] Christos H. Papadimitriou y Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall,1982

[25] Vladimir Kolmogorov y Ramin Zabih. Multi-camera Scene Reconstruction via Graph Cuts. European Conference on Computer Vision, 2002

[26] Peter N. Belhumeur. A Bayesian Approach to Binocular Stereopsis. International Journal of Computer Vision, 1996

[27] Federico Lecumberry Ruvertoni, "*Cálculo de Disparidad y Segmentación de Objetos en Secuencias de Video*", Tesis de Maestría en Ingeniería Eléctrica, 2005

[28] C. Lawrence Zitnick, Takeo Kanade, "*A Cooperative Algorithm for Stereo Matching and Occlusion Detection*", The Robotics Institute,Carnegie Mellon University ,1999.

APÉNDICE 1

En éste apartado se describen los distintos métodos que se utilizan para calcular la matriz de transformación asociada a cada homografía (H). En especial se presenta el algoritmo DLT, el algoritmo mas extendido en el cálculo de homografías.

8.1 ALGORITMO DE LA TRANSFORMACIÓN LINEAL DIRECTA (DLT)

Comenzaremos estudiando un algoritmo lineal simple para determinar H dado un conjunto de cuatro puntos en correspondencia. El primer punto a observar es que la ecuación x'= Hx esta definida para coordenadas homogéneas y por tanto la igualdad hay que interpretarla como: "ambos miembros son iguales salvo una constante de proporcionalidad". Es decir las coordenadas de los vectores de x' y Hx pueden diferir en una constante de proporcionalidad. Para evitar esta constante de proporcionalidad expresaremos la ecuación de la siguiente forma equivalente que nos permitirá estimar H, $x_i' \times Hx_i = 0$ para todo i.

Si notamos por h^{jT} la j-ésima fila de la matriz H, podemos escribir

$$Hx_i = \begin{pmatrix} h^{1T}x_i \\ h^{2T}x_i \\ h^{3T}x_i \end{pmatrix} \qquad \text{Ec. 35}$$

escribiendo $x_i'=(x_i',y_i',w_i')^T$, el producto vectorial puede expresarse como

$$Hx_i = \begin{pmatrix} y'_i h^{3T}x_i - w'_i h^{2T}x_i \\ w'_i h^{1T}x_i - x'_i h^{3T}x_i \\ x'_i h^{2T}x_i - y'_i h^{1T}x_i \end{pmatrix} \qquad \text{Ec. 36}$$

igualando a cero esta ecuación obtenemos un conjunto de tres ecuaciones en los parámetros de H que se puede expresar de la siguiente manera

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \qquad \text{Ec. 37}$$

Estas ecuaciones tiene la forma $A_i h=0$, donde A es una matriz 3x9 y h es un 9-vector, $h=(h^{1T}, h^{2T}, h^{3T})^T$, compuesto por los elementos de H.

A la vista de estas ecuaciones cabe hacer los siguientes comentarios:

1. La ecuación $A_i h=0$ es lineal en las incógnitas h pero los elementos de A_i son cuadráticos en las coordenadas de los puntos conocidos.

2. Aunque tenemos tres ecuaciones en el sistema anterior tan solo dos son linealmente independientes, ya que la tercera fila se puede obtener fácilmente como combinación lineal de las otras dos. Por tanto cada punto tan solo proporciona dos ecuaciones en las entradas de H. Usualmente la tercera ecuación no se suele usar en la estimación, por tanto el sistema de ecuaciones para cada correspondencia se reduce a:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \qquad \text{Ec. 38}$$

y por tanto en el sistema $A_i h=0$ la matriz A será 2x9. Si alguno de los puntos x_i es un punto ideal $w_i=0$, estas dos ecuaciones se reducen a solo una, pero el sistema de tres ecuaciones sigue conservando dos linealmente independientes. Por tanto una regla de actuación que nos libera de tener que verificar la posibilidad de puntos ideales pasa por no suprimir la ultima ecuación del sistema de 3 ecuaciones en el proceso de estimación.

3. Las ecuaciones se verifican para cualquier representación de las coordenadas homogéneas del punto x_i . En particular podemos elegir $w_i=1$, lo que significa que las coordenadas (x_i, y_i') son las coordenadas medidas sobre las imágenes. Otras opciones también son posibles.

Calculo de H

Ya que cada correspondencia de puntos da lugar a dos ecuaciones independientes, fijadas cuatro correspondencias podemos construir un sistema de ecuaciones Ah=0 de 8 ecuaciones con 9 incógnitas. La solución trivial del sistema de ecuaciones h=0 no es de interés para nosotros, así que hay que buscar las otras soluciones del sistema. La matriz A tendrá dimensiones 12x9 si se consideran tres ecuaciones por punto, o 8x9 si sólo se consideran dos ecuaciones, pero en ambos casos el rango de la matriz A es 8, por tanto existe un vector solución del sistema dado por el vector del núcleo de la aplicación lineal definida por la matriz A. Es conocido que dicho vector sólo puede ser determinado salvo escala, pero por otro lado sabemos que la matriz H y por tanto el vector h esta definido salvo escala. Con objeto de fijar una escala para los cálculos sobre el vector h se puede tomar la condición sobre su norma ||h||=1.

Solución sobredeterminada

Si disponemos de más de cuatro puntos en correspondencia el sistema de ecuaciones Ah=0 que nos proporciona estará sobredeterminado. Si la posición de los puntos es exacta entonces el rango de la matriz A seguiría siendo 8 y existiría una

solución como en el caso básico, en cambio si los puntos contienen ruido en sus coordenadas el rango de A podría ser superior y por tanto la única solución al sistema sería la solución trivial h=0. En este caso, en lugar de pedir una solución exacta se intentaría obtener una solución aproximada, es decir un vector h que minimiza una función de coste conveniente. La cuestión fundamental que se presenta es, ¿que función debería ser minimizada?

Evidentemente para impedir que la función de optimización alcance la solución trivial habrá que fijar que ||h||=1. El valor de la norma no tiene interés ya que sabemos que el vector h esta definido salvo una escala.

Dado que no existe una solución exacta a Ah=0 parece razonable que en su lugar minimicemos la norma ||Ah|| sujeta a la condición ||h||=1. Este problema es equivalente al problema de minimizar respecto h el cociente ||Ah||/||h||. Es conocido de la teoría de matrices que el óptimo de este problema está dado por el vector singular unidad de la matriz A asociado al menor valor singular de la misma.

ALGORITMO-DLT

Objetivo: Dado n≥4 parejas de puntos en correspondencia $x_i \leftarrow\rightarrow x_i'$, determinar la homografía 2D, H tal que $x_i' = Hx_i$.

Pasos:

1. Para cada correspondencia $x_i \leftarrow\rightarrow x_i'$ calcular la matriz A_i con dos ecuaciones (en general solo serán necesarias dos).

2. Construir la matriz A (2nx9) a partir de las n matrices A_i

3. Obtener la descomposición en valores singulares de la matriz A. El vector singular unidad asociado al menor valor singular será la solución h.

4. La matriz H se determina a partir de h por filas.

Solución no-homogénea

Una alternativa a calcular h directamente como un vector homogéneo es convertir el conjunto de ecuaciones Ah=0 en un conjunto no-homogéneo imponiendo la condición h_j=1 para algún valor de j. Esta solución se justifica ya que sabemos que el vector solución esta definido salvo un factor de escala, y esta puede elegirse de manera que h_j=1. En este caso el sistema de ecuaciones Ah=0 para cuatro puntos se convierte en un sistema determinado, de 8 ecuaciones con 8 incógnitas, estando el vector del término independiente formado por los coeficientes de h_j. Este sistema puede resolverse usando las técnicas usuales de resolución de sistemas de ecuaciones.

Sin embargo, la elección de h_j=1, impone que el vector solución tiene que tener obligatoriamente en la coordenada j-ésima un valor distinto de cero lo cual dado lo arbitrario de la elección de j no puede garantizarse a-priori. En aquellos en que el verdadero valor sea h_j=0 este método no podrá encontrar la verdadera solución. Evidentemente, este método producirá solucione inestables sí el verdadero valor de h_j esta cercano a cero. En general este método no es recomendable por estos problemas.

Normalmente se supone que h_{33}=1, pero puede verse que el caso h_{33}=0 no es tan extraño como puede pensarse. En el caso de que el origen de coordenadas de la imagen este en un punto que sea imagen de un punto del infinito de la escena, tendremos h_{33}=0. En muchas imágenes la línea del horizonte (imagen de la línea del infinito) se sitúa en medio de la imagen y por tanto con alta probabilidad de que el origen de coordenadas caiga sobre ella.

8.2 FUNCIONES DE COSTO

Vamos a estudiar las distintas funciones de costo que pueden usarse para determinar h en el caso sobredeterminado.

8.2.1 DISTANCIA ALGEBRAICA

El algoritmo DLT minimiza la norma ||Ah||. El vector ε = Ah se denomina el vector residual y es por tanto la norma de este vector lo que es minimizado. Las componentes de este vector se generan a partir de las ecuaciones de correspondencia de cada pareja de puntos $x_i \leftarrow\rightarrow x_i'$. Cada correspondencia genera un vector de error parcial ε_i , cuya unión genera el vector ε. Este vector ε_i es el vector de error algebraico asociado a la correspondencia $x_i \leftarrow\rightarrow x_i'$ y la homografía H. La norma de este vector es un escalar que se denomina la *distancia algebraica.*

$$d_{alg}(x', Hx)^2 = ||\varepsilon_i||^2 = \left\| \begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & 0^T \end{bmatrix} h \right\|^2 \quad \text{Ec. 39}$$

Dado un conjunto de correspondencias, la cantidad ε = Ah es el vector de error algebraico para el conjunto completo y puede verse que

$$||\varepsilon||^2 = ||Ah||^2 = \sum_i ||\varepsilon_i||^2 \quad \text{Ec. 40}$$

En la literatura se ha mostrado que estos métodos usados de forma directa a veces no producen los resultados que intuitivamente cabe esperar. Sin embargo, si se normalizan adecuadamente las coordenadas de los puntos, este método produce resultados bastante buenos. Estudiaremos el proceso de normalización más adelante.

8.2.2 DISTANCIA GEOMÉTRICA

A continuación estudiaremos funciones de error alternativas basadas en medidas de distancias geométricas en la imagen y la minimización de la diferencia entre las coordenadas medidas y estimadas sobre la imagen.

El vector x notará ahora las coordenadas medidas en la imagen, $\hat{\bar{X}}$ representará los valores estimados y $\bar{X}$ representará los verdaderos valores de los puntos.

Error en una imagen. Comenzaremos suponiendo que solamente existe error en la segunda imagen, es decir las coordenadas de los puntos de la primera imagen están medidos sin error (en la practica supondrá que se han medido de forma muy precisa). En este caso lo que se desea minimizar es lo que se denomina el *error de transferencia*. Esto es la distancia euclídea en la segunda imagen entre las coordenadas medidas para el punto x' y las coordenadas del punto imagen por la homografía H de su correspondiente en la primera imagen Hx. Si notamos por d(x, x') la distancia euclídea entre las coordenadas no-homogéneas de dos puntos, el error de transferencia para un conjunto de correspondencias es

$$\sum_i d(x'_i, H\bar{x}_i)^2 \qquad \text{Ec. 41}$$

La homografía estimada $\hat{H}$ es aquella que minimiza el error de transferencia.

Error de transferencia simétrico: En el caso más realista donde existen errores de medidas en ambas imágenes, es preferible que los errores sean minimizados en ambas imágenes y no solamente en una. Una forma de construir una función de error más satisfactoria es considerar la transformación hacia delante H y hacia atrás H^{-1} y sumar los errores geométricos correspondientes a cada una de estas dos transformaciones. Por tanto el error será

$$\sum_i d(x_i, H^{-1}x'_i)^2 + d(x'_i, H\bar{x}_i)^2 \qquad \text{Ec. 42}$$

El primer término de esta suma será el error en la primera imagen y el segundo término el error en la segunda imagen. De nuevo, la homografía estimada Ĥ es aquella que minimiza el error de transferencia.

Error de retroproyección-ambas imágenes: Un método alternativo de cuantificar el error en cada una de las dos imágenes conlleva la estimación de una corrección de las coordenadas de los puntos para cada correspondencia. Uno puede preguntarse cuanto es preciso corregir las coordenadas de los puntos en ambas imágenes para obtener un conjunto perfectamente emparejado de puntos imagen. Este razonamiento se puede comparar con el realizado en el caso de error en solo una imagen, ya que este caso es un caso particular, para solo una imagen, del que queremos estudiar.

Ahora lo que estamos buscando es una homografía Ĥ y parejas de puntos perfectamente emparejadas que minimicen la función de error total

$$\sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}_i)^2 \qquad \text{Ec.43}$$

Sujeto a $\hat{x}'_i = \hat{H}\hat{x}_i \quad para\ todo\ i$

Minimizar esta función de costo conlleva determinar tanto Ĥ como el conjunto de puntos subsidiarios $\{\hat{x}_i\}$ y $\{\hat{x}'_i\}$

Esta función de error de retroproyección se compara con la función de error simétrica de la siguiente figura.

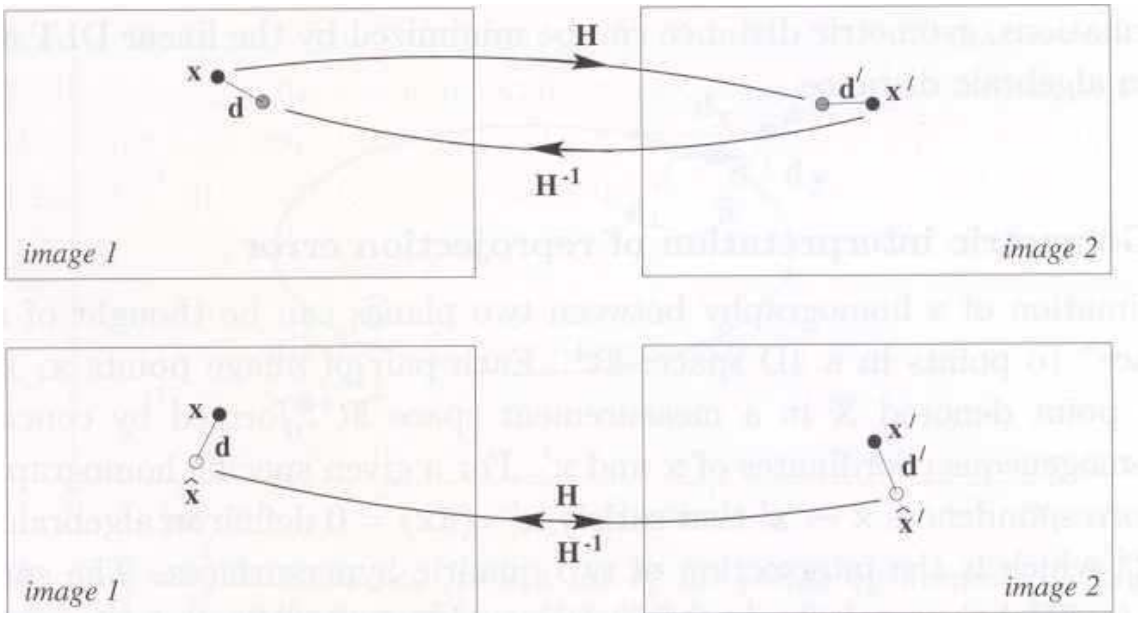

FIGURA 108: LA FIGURA MUESTRA UNA COMPARACIÓN ENTRE EL ERROR DE TRANSFERENCIA SIMÉTRICO (ARRIBA) Y EL ERROR DE RETROPROYECCIÓN (ABAJO) EN LA ESTIMACIÓN DE UNA HOMOGRAFÍA

Esta estimación modela también la situación en la que los puntos en correspondencia son imagen de puntos en un mundo plano. En este caso deseamos estimar un punto sobre un mundo plano $\ddot{X}_i$ a partir de $x_i \leftarrow \rightarrow x_i'$ que es entonces retroproyectado a la estimación de la correspondencia de puntos perfectamente emparejados $\hat{x}_i \leftarrow \rightarrow \hat{x}'_i$

8.2.3 TRANSFORMACIÓN DE NORMALIZACIÓN

Una solución al problema de ausencia de invariancia del algoritmo DLT es aplicar una transformación de normalización a las imágenes antes de aplicar el algoritmo DLT de estimación de H. Esta normalización equilibrará el efecto de una elección arbitraría del sistema de referencia sobre la imágenes y por tanto obtendremos más precisión en la estimación. Además y como consecuencia de elegir un sistema de coordenadas canónico para los datos, tiene el efecto añadido de hacer que el algoritmo DLT sea invariante ante transformaciones de semejanza.

Como un primer paso en el proceso de normalización, se realiza una traslación del origen de coordenadas en cada imagen (normalmente distinta en cada una) de manera que el centroide de los puntos se convierta en el nuevo origen de coordenadas. El segundo paso es realizar un cambio de escala en las coordenadas de la imagen con el objetivo de que todos los ejes coordenados tengan en promedio la misma magnitud. Realmente más que escoger un valor de escala diferente para cada eje, se elige un único valor común para todos ellos. Para ello se elige escalar las coordenadas de manera que la distancia promedio de un punto de la imagen al origen sea igual a sqrt(2), esto significa que el punto promedio es igual al $(1,1,1)^T$. En resumen la transformación es como sigue

1. Los puntos son trasladados de manera que su centroide sea el nuevo origen
2. Los puntos son escalados de manera que su distancia promedio desde el nuevo origen sea sqrt(2)
3. Esta transformación se aplica a ambas imágenes independientemente.

Por tanto el proceso de normalización no es otra cosa que la búsqueda de dos matrices 3x3 T y T' tal que $\hat{x} = Tx$ y $\hat{x}' = T'x'$ y que hagan que los puntos $\hat{x}$ y $\hat{x}'$ verifiquen las condiciones anteriores. Las siguientes transformaciones sobre los valores de x y x´ permiten obtener valores con las condiciones antes establecidas

$$\hat{x}' = \frac{1}{n}\sum_{i=1}^{n} x'_i \qquad \text{Ec. 44}$$

$$\hat{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{Ec. 45}$$

$$\sigma_x = \frac{1}{n\sqrt{2}} \sum_{i=1}^{n} \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2}$$ Ec. 46

$$\sigma_{M'} = \frac{1}{n\sqrt{2}} \sum_{i=1}^{n} \sqrt{(x'_i - \hat{x'})^2 + (y'_i - \hat{y'})^2}$$ Ec. 47

$$\tilde{x} = (x - \hat{x})/\sigma_x$$ Ec. 48

$$\tilde{x'} = (x' - \hat{x'})/\sigma_{x'}$$ Ec. 49

Los elementos de las matrices T y T' se deducen fácilmente a partir de las ecuaciones de la transformación.

Esta normalización de las coordenadas debe ser un paso obligado en la aplicación del algoritmo DLT. La versión definitiva del algoritmo será la siguiente:

8.2.4 ALGORITMO DLT (FINAL)

1. Normalización de las coordenadas en cada una de las imágenes de forma independiente

2. Aplicar el algoritmo DLT a los nuevos puntos y estimar $\hat{\mathbf{H}}$

3. Estimar **H** invirtiendo la transformación de normalización. $\mathbf{H} = \mathbf{T}'^{-1} \hat{\mathbf{H}} \mathbf{T}$

PRESUPUESTO

1) Ejecución Material

· Compra de ordenador personal (Software incluido)..... 2.000 €
· Alquiler de impresora láser durante 6 meses..... 60 €
· Material de oficina.....200 €
· Total de ejecución material.....2.260 €

2) Gastos generales

· 16 % sobre Ejecución Material..... 362 €

3) Beneficio Industrial

· 6 % sobre Ejecución Material..... 136 €

4) Honorarios Proyecto

· 1500 horas a 15 € / hora..... 22500 €

5) Material fungible

· Gastos de impresión..... 50 €
· Encuadernación..... 150 €

6) Subtotal del presupuesto

· Subtotal Presupuesto..... 25458 €

7) I.V.A. aplicable

· 16% Subtotal Presupuesto..... 4074 €

8) Total presupuesto

· Total Presupuesto ..... 29532 €

Madrid, Septiembre de 2010
El Ingeniero Jefe de Proyecto
Fdo.: David Otero García
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de una fusión de secuencias de vídeo de alta velocidad procedentes de cámaras desplazadas espacial o temporalmente. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

CONDICIONES GENERALES

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda

exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

CONDICIONES PARTICULARES

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.