

Cálculo de disparidad en imágenes estéreo, una comparación

Federico Lecumberry

fefo@fing.edu.uy

IIE - Fac. de Ingeniería, Universidad de la República, Uruguay

III Workshop de Computación Gráfica, Imágenes y Visualización

Resumen

La estimación de la profundidad de los objetos en una escena a partir de un par estéreo es fuente para muchas aplicaciones en el mundo de computación gráfica, *Image Based Rendering*, realidad virtual, navegación de robots, vigilancia, etc. Los métodos para la estimación de la profundidad mediante el cálculo de disparidad han variado mucho en las décadas que el problema está planteado. Recientes comparaciones [1] dan a los métodos basados en Corte de Grafos como los que generan resultados con menor cantidad de errores. En este trabajo se comparan dos métodos de cálculo de disparidad, uno basado en Corte de Grafos y otro en Programación Dinámica en varios experimentos. Se presenta una introducción al problema del cálculo de disparidad y la reconstrucción euclídea de la escena.

Keywords: Computación Gráfica, Procesamiento de Señales, Cálculo de Disparidad.

1. Introducción

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 al observador. El cerebro es capaz de interpretar esa diferencia y reconstruir la estructura de la escena que ve el observador. Según Marr y Poggio [2] 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. Se llama visión estéreo a esta 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, y más cercana al problema computacional, es 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 (ver figura 1(a)). De esta forma se recupera una imagen *densa* de profundidades para cada uno de los puntos de la escena proyectados en ambas imágenes. La figura 5 muestra un par de imágenes de una escena tomadas con un pequeño desplazamiento horizontal como serían vistas por el ojo izquierdo y derecho, respectivamente.

En las últimas tres (casi cuatro) décadas el tema de la visión estéreo ha sido abordado por la comunidad de Computer Vision. Existen muchos trabajos en el tema de cálculo de disparidad, con

soluciones ajustadas a diferentes requerimientos y utilizando distintas técnicas. Dentro de las técnicas más utilizadas se encuentran las que se basan en Programación Dinámica (DP) [2, 3, 4, 5, 6] con varias décadas de uso, y más recientemente las basadas en Corte de Grafos (GC) [7, 8, 9]. Existen publicaciones que clasifican [10] y comparan [1] estos diversos métodos, dando los mejores resultados a los métodos basados en Corte de Grafos por lo menos desde el punto de vista del número de errores cometidos.

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. [11, 12, 13]

En la sección 2 se hace una presentación geométrica de la relación entre la disparidad y la profundidad de los objetos en la escena, y del uso de la misma para la reconstrucción euclídea de la estructura de una escena tridimensional. En la sección 2.1 se presenta la estructura clásica y las consideraciones de los algoritmos estéreo aplicados al problema del cálculo de disparidad. En la sección 3 se presentan los algoritmos comparados. En la sección 4 se presentan los experimentos realizados y la discusión de los resultados. En la sección 5 se presentan las conclusiones del trabajo.

2. La 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 1(a). 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, $\overrightarrow{O_R O_R} \parallel \overrightarrow{O_L O_L}$. Ambas cámaras tienen la misma distancia focal, f , con centros O_L y O_R separados una distancia B , llamada línea base (*baseline*), de forma que las imágenes que se forman, I_L e I_R , estén en planos paralelos. De esta manera la línea base es paralela a la coordenada \vec{x} de las imágenes. Con el modelo *pinhole* considerado, un punto en el espacio tridimensional P , con coordenadas homogéneas $(X, Y, Z, 1)^T$, se proyecta en cada una de las imágenes bidimensionales en los puntos p_L y p_R , con coordenadas homogéneas $(x_L, y_L, 1)^T$ y $(x_R, y_R, 1)^T$, respectivamente.

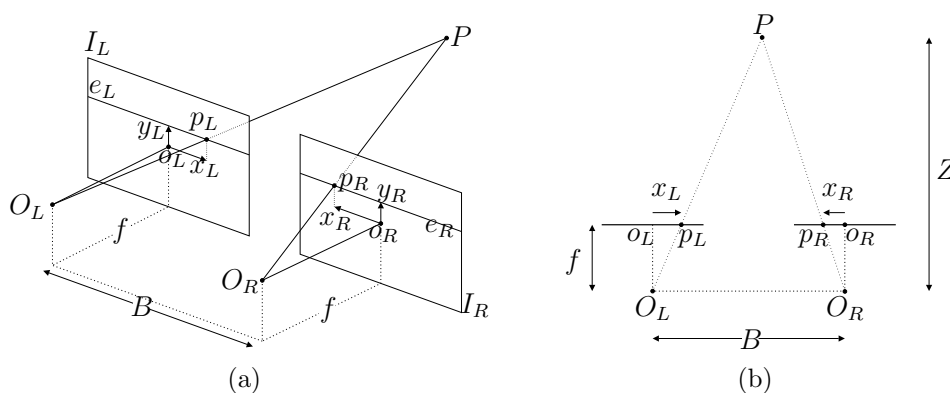


Figura 1: (a) Configuración de las cámaras del par estéreo. (b) Relación geométrica entre los parámetros del par estéreo para obtener la profundidad Z a partir de la disparidad d .

El plano que contiene a los puntos P , O_L y O_R , interseca a las imágenes en dos rectas e_L y e_R . Dada la configuración del par estéreo, éstas son *rectas epipolares* entre sí, o sea, un punto, p_L , en

la recta e_L de la imagen I_L tiene su correspondiente en algún punto de la recta e_R . Esto reduce la búsqueda del correspondiente de p_L de toda la imagen I_R a la recta e_R . En la figura 1(b) podemos ver cómo 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 .

La disparidad es la diferencia en las coordenadas horizontales de los puntos p_L y p_R , o sea, $d = x_L - x_R$. Dependiendo el sistema de referencia utilizado en las imágenes, la definición puede cambiar de forma que el signo sea siempre positivo. Las coordenadas de p_L y p_R quedan relacionadas mediante $x_L = x_R + d$ y $y_L = y_R$. Por semejanza entre los triángulos $PO_L O_R$, $p_R o_R O_R$ y $p_L o_L O_L$ se llega a la relación entre d y Z :

$$Zd = fB \quad (1)$$

De esta formase puede recuperar, a menos de una constante de escala, la profundidad de cada pixel en cada una de las imágenes a partir de la disparidad calculada. La relación de proporcionalidad inversa planteada en (1) es fácilmente verificable observando alternadamente las imágenes izquierda y derecha, y notando que los objetos mas cercanos a la cámara –menor Z – tienen mayor desplazamiento relativo –mayor d – en las imágenes (ver figura 5).

La geometría proyectiva brinda las herramientas para trabajar analíticamente con estas relaciones geométricas. El uso de las coordenadas homogéneas permite considerar el modelo *pinhole* de la cámara como una transformación lineal entre la escena y el plano de la imagen mediante la matriz de cámara, \mathbf{IP}_C

$$kp = \mathbf{IP}_C P \Rightarrow (kx, ky, k)^T = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} (X, Y, Z, 1)^T \quad (2)$$

donde la matriz que se muestra en la ecuación (2) es la forma más sencilla para la matriz de cámara. Esto es válido para todo $k \neq 0$ y permite hallar la relaciones entre las coordenadas de P y p ,

$$Zx = fX \quad Zy = fY \quad (3)$$

Estas ecuaciones pueden verificarse geoméricamente en la figura 1(a) que, junto con la ecuación (1) y los parámetros intrínsecos de la cámara (f), y la configuración geométrica del par estéreo (B) hacen posible recuperar la profundidad de cada punto de la imagen, y por lo tanto la estructura euclídea de la escena.

2.1. Algoritmos para el cálculo de disparidad

El objetivo de un algoritmo estéreo de cálculo de disparidad es obtener la profundidad en *todos* los pixeles de las imágenes del par estéreo. Hallando las parejas de puntos correspondientes que son proyección del mismo punto del espacio, para todos los puntos en cada una de las imágenes. De esta forma se recupera un mapa *denso* de disparidades.

Los algoritmos de cálculo de disparidad asumen, por simplicidad, que las imágenes con que se trabaja están rectificadas, es decir que los planos de las imágenes en cada cámara son paralelos entre sí, y paralelos a la dirección en la cual existe el desplazamiento entre las imágenes. Normalmente esto se garantiza en la forma de adquisición de las imágenes [14]. De esta forma el correspondiente de un punto de la en la fila y_L de la imagen izquierda se encuentra en la fila $y_R = y_L$ de la imagen derecha.

Para hallar los correspondientes se imponen restricciones (hipótesis) basadas en propiedades físicas y geométricas razonables de los objetos y superficies presentes en la escena, y su relación. Las restricciones comúnmente mencionadas en la literatura son: (i) la *restricción epipolar* [15] (el correspondiente de un punto en una imagen debe estar en la recta epipolar del punto en la otra imagen), (ii) la *restricción de orden* (si la proyección del objeto Q está a la izquierda de la proyección del objeto P en la imagen izquierda, entonces la proyección de Q debe estar a la izquierda de la proyección de P en

la imagen derecha), (iii) la *restricción de unicidad* [2] (cada punto de una imagen puede tener no más de un correspondiente en la otra imagen) y (vi) la *restricción de semejanza* (las características de los puntos en las imágenes (intensidad o color, etc.) no debe cambiar mucho).

Otro fenómeno a tener en cuenta en el planteo de la visión estéreo y que influye en el proceso de hallar los puntos correspondientes, son las *oclusiones*, o regiones que se ven en una imagen y que no se ven en la otra imagen por estar tapadas por un objeto. Las oclusiones siempre implican una discontinuidad en la profundidad de la escena y son fuente de errores en muchos algoritmos estéreo. Sin embargo estas regiones ocultas pueden ser usadas para la recuperación de la estructura de la escena y dan información fundamental de ésta.

3. Algoritmos comparados

Para la comparación de los algoritmos de cálculo de disparidad se tomó un representante de los algoritmos basados en Programación Dinámica y uno basado en Cortes de Grafo. El primero es el algoritmo de Bobick e Intille [3] donde se hace uso de muchas de las herramientas más utilizadas en los algoritmos basados en DP. El segundo es el algoritmo de Kolmogorov y Zabih [7] uno de los algoritmos con mejores resultados en el cálculo de disparidad [1] basado en GC.

3.1. Algoritmo de Bobick e Intille -DP-

Bobick e Intille presentan [3] un algoritmo que modela las oclusiones y las integra en el proceso de cálculo de un costo mínimo mediante el uso de Programación Dinámica. Introducen el concepto de *Ground Control Points* y utilizan una *imagen del espacio de disparidad* (DSI). El algoritmo se ejecuta para cada una de las filas (llamadas *scanlines*) por separado y no se introduce información de las filas adyacentes. Esto presenta desventajas pues no se fuerza una coherencia entre las *scanlines* adyacentes en el proceso de cálculo de una *scanline* en particular. Para cada pareja de *scanlines* correspondientes, se crea la «imagen del espacio de disparidad» donde en cada pixel (x, d) de esta «imagen» se coloca una medida de la diferencia entre el pixel x de I_L y el pixel $x + d$ de I_R . Esta «imagen» tiene la misma cantidad de columnas de ancho que las imágenes originales y el rango de disparidades buscado ($d_{max} - d_{min}$) filas.

Encontrar un recorrido de un extremo a otro del DSI implica determinar la disparidad de cada punto en la *scanline*. El costo de este camino es la suma de los valores del DSI en los puntos del camino. El camino de menor costo a través del DSI se calcula usando Programación Dinámica. En este camino se definen ciertas restricciones en cómo se recorre pues no todos los caminos son válidos debido a las oclusiones en la escena. Cuando el algoritmo no logra encontrar una pareja de correspondientes, supone que el punto está oculto, y asigna un costo de oclusión que se suma al valor del DSI en el pixel, aumentando el costo del camino. Cuando no existe una oclusión el camino es horizontal en el DSI.,



Figura 2: Izquierda: Imagen del espacio de disparidad para un caso real (ecualizado para visualización). Derecha: En azul el camino hallado, en rojo los GCP, en gris las regiones prohibidas para el camino.

dando una disparidad constante, correspondiente a un plano frente a la cámara paralelo al plano de la imagen, donde la profundidad será la misma para todos los puntos del plano. En las oclusiones existe un cambio de profundidad y el camino debe «saltar» en la dirección correcta. El valor del costo de oclusión es un parámetro fundamental del algoritmo. Un valor alto hará que no se elija nunca la opción de una oclusión devolviendo un mapa de disparidad constante. Por otro lado, un valor bajo, hará que el camino obtenido sea muy *ruidoso* y seguramente erróneo.

Una de las características principales de este algoritmo es el uso de lo que los autores denominan *Ground Control Points* (GCP). Estos puntos se definen como puntos relevantes del DSI en los que puede asegurarse la correspondencia; definir el punto del DSI (x_G, d_G) como GCP implica afirmar que los puntos con coordenadas $x_L = x_G$ y $x_R = (x_G + d)$, se corresponden. Los GCP ayudan a encontrar el camino óptimo a través del DSI en presencia de oclusiones grandes, forzando el camino óptimo a través de los mismos. En la figura 2 se muestra el camino a través de un DSI teniendo en cuenta los GCP. Ver cómo se conduce el camino hacia el GCP.

Los mapas de disparidad que se obtienen con este algoritmo son aproximados a la solución real. En la figura 3 se puede ver los resultados variando el costo de oclusión. La solución es sensible al valor del costo de oclusión utilizado. El uso de los GCP colabora en este sentido, al garantizar algunos *matcheos* correctos, y ayudan a obtener una mejora local de la solución. De las desventajas más notorias de este algoritmo, y de todos los algoritmos que se basan en búsquedas intra-scanline, sin incorporar información de *scanlines* adyacentes (inter-scanline), es el «rayado» horizontal que se ve en los mapas de disparidad calculados. Este rayado se origina al tener distintas soluciones para *scanlines* consecutivas, debido, principalmente a la propagación de un error.



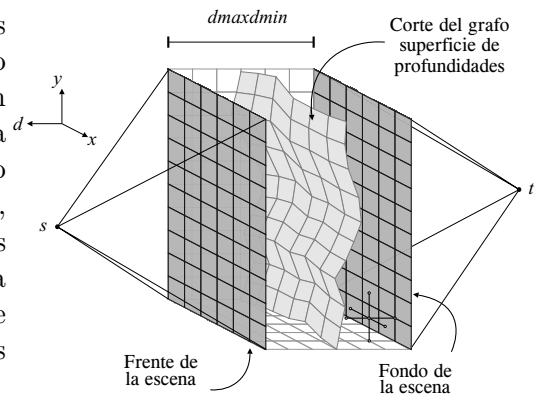
Figura 3: Mapas de disparidad para *tsukuba* obtenidos con el algoritmo de Bobick e Intille. De izquierda a derecha aumentando el costo de oclusión. La disparidad real se muestra en la figura 5.

3.2. Algoritmo de Kolmogorov y Zabih -GC-

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 quienes presentan [8] un algoritmo donde construyen el grafo con un nodo por cada pixel de la imagen en cada posible valor de disparidad, o sea $(d_{max} - d_{min}) \times m \times n$, más los dos nodos terminales, s y t , donde m y n son el número de filas y de columnas de las imágenes. Esta estructura de grafo representa alguna expresión de energía [8, 7, 9], el corte mínimo del grafo que la minimiza corresponde con la superficie de profundidades de la escena.

Kolmogorov y Zabih presentan [7] un algoritmo que se basa en las técnicas de Corte de Grafos para hallar la correspondencia entre los puntos de las imágenes, imponiendo la restricción de unicidad. Para resolver de forma eficiente el corte del grafo presentan dos algoritmos con los que obtienen tiempos mucho menores que con otros métodos de corte de grafos. La energía que se plantea tiene la expresión

$$\begin{aligned}
 E(L) &= E_{data}(L) + E_{oclu}(L) + E_{smooth}(L) = \\
 &= \sum (I(p) - I(q))^2 + \sum C_p T(N_C = 0) + \sum_{\{p,q\} \in \mathcal{N}} D_{p,q} T(L_p \neq L_q)
 \end{aligned} \tag{4}$$



donde L es una configuración de posibles etiquetas a colocar en cada punto del grafo (que corresponde con parejas de píxeles en las imágenes). Para el caso de cálculo de disparidad, las etiquetas son: los posibles valores de disparidad, $d \in [d_{min}, d_{max}]$, y la etiqueta de oculto. El término $E_{data}(L)$ mide la semejanza entre puntos correspondientes, basado en la diferencia en los niveles de gris de los mismos, donde p y q son los puntos correspondientes. $E_{oclu}(L)$ añade un costo fijo, C_p , al asignar la etiqueta de oculto a un punto; el cual se determina cuando no se le puede asignar otro punto correspondiente (número de correspondientes $N_C = 0$) donde $T(\cdot)$ vale 1 si el argumento es verdadero o 0 si es falso. Finalmente, $E_{smooth}(L)$ es un término de suavidad que intenta asignar una disparidad similar a los puntos cercanos; imponiendo una coherencia en todas las direcciones, no sólo en la de la *scanline*. $D_{p,q}$ es un potencial de penalización cuando una pareja de puntos $\{p, q\}$ en un vecindario \mathcal{N} tiene diferentes disparidades.



Figura 4: Mapas de disparidad para *tsukuba* obtenidos con el algoritmo de Kolmogorov y Zabih. De izquierda a derecha aumentando λ . En rojo los puntos marcados como ocultos.

En la implementación que proponen los autores, todas las componentes de la energía (3.2) son configuradas a partir de un único parámetro λ , fijando otros por resultados experimentales. Al término $E_{smooth}(\cdot)$ se le asigna un valor proporcional a λ . El costo de oclusión, C_p , también es proporcional a λ . En la figura 4 se ven los mapas de disparidad estimados variando el parámetro λ . Los resultados que se obtienen son mucho más próximos al real que con DP.

4. Experimentos y discusión

Se realizó la comparación de los dos métodos de cálculo de disparidad (los llamaremos DP y GC respectivamente), con las imágenes de prueba que se muestran en la figura 3.

El primer experimento que se realiza es añadir ruido gaussiano aditivo con diferentes varianzas a las imágenes estéreo. Los resultados se obtienen variando un parámetro en cada uno de ellos. Para DP se hizo variar el *costo de oclusión*. Para GC, se hace variar el parámetro λ con los ajustes que dan los autores.

Para los experimentos se usaron imágenes estéreo obtenidas de bases públicas, en niveles de grises (8 bits), para las cuáles se conoce la disparidad real. El rango de disparidad de las imágenes seleccionadas es de 16 píxeles. Las imágenes que se utilizaron son conocidas como *tsukuba* y *corridor*. La primera es una imagen de una escena real, donde se pueden apreciar varios objetos relativamente planos dada la distancia a la que se encuentran de la cámara. La disparidad real fue obtenida manualmente por los autores. La mayor dificultad se plantea en las oclusiones que se dan entre los objetos; y en el rostro de la cabeza. La segunda es una escena artificial, generada por computadora, en la cual se ve un pasillo con varios objetos en el piso y las paredes. La profundidad de la escena varía de forma continua y rápida en las paredes, piso y techo del corredor provocando que la mayoría de los algoritmos no puedan seguir esta variación, generando un mapa de disparidad cuantificado en forma gruesa, provocando errores importantes.

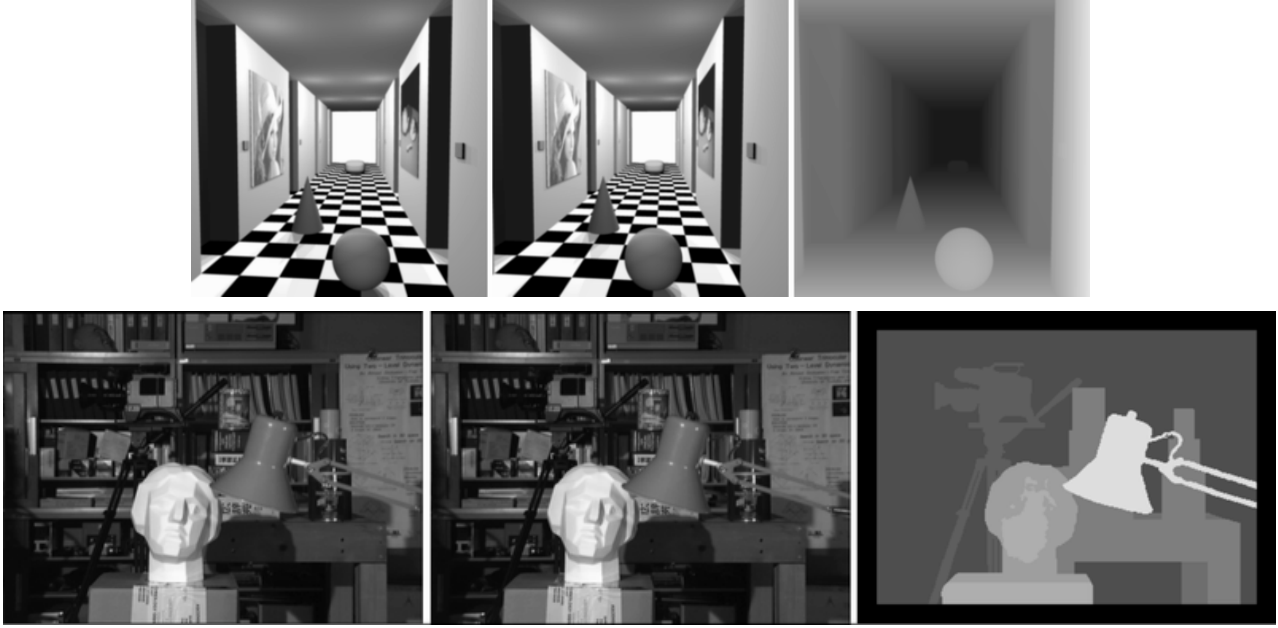


Figura 5: Imágenes izquierda y derecha de escenas utilizadas en las pruebas, y mapa de disparidad real para la imagen izquierda de cada escena. Arriba: *corridor*. Abajo: *tsukuba*.

Las medidas que se tomaron fueron: la cantidad de puntos donde fue mal calculada la disparidad, B , y una medida de la potencia del error de disparidad, RMS . Estas medidas fueron planteadas por Scharstein y Szeliski [1],

$$B = \frac{1}{N} \sum_{x,y} |d_C(x,y) - d_R(x,y)| > \Delta_d \quad RMS = \sqrt{\frac{1}{N} \sum_{x,y} (d_C(x,y) - d_R(x,y))^2} \quad (5)$$

donde $d_C(x,y)$ es la disparidad calculada, $d_R(x,y)$ es la disparidad real en el punto (x,y) , Δ_d es una tolerancia al error en la disparidad y N es el número de puntos donde se calcula la disparidad. Fue tomado $\Delta_d = 1$ para las gráficas que se presentan.

Para las imágenes de *corridor* además se hizo el experimento de estudiar las soluciones que presentan los algoritmos a lo largo del corredor, en la precisión de la variación de la disparidad.

Los mapas de disparidad generados con DP se muestran en las figuras 8(b) y 10. Al aumentar la potencia del ruido introducido en las imágenes del par estéreo, la potencia del error aumenta, al igual que el número de puntos con disparidad mal calculada. Este comportamiento se ve claramente en la figuras 10 y 6. Es interesante el comportamiento al variar el costo de oclusión. En la figura 6(a) se observa que la potencia del error disminuye al aumentar el costo de oclusión. La explicación se debe a que al aumentar el costo de oclusión la capacidad de cambio de la disparidad es menor (el costo de «saltar» en el camino del DSI es mayor), por lo que se adapta más a regiones suficientemente planas y paralelas al plano de la imagen; como las que se ven en *tsukuba*. Por otro lado, en *corridor*, los planos tienen una pendiente importante. En la figura 6 vemos que, tanto RMS como B , no tiene el comportamiento anterior para potencias de ruido bajas; en este caso, tiene un aumento pequeño (una variación relativa menor del 15%). Para potencias de ruido mayores el comportamiento es el mismo que en el caso anterior.

Estos resultados se pueden observar en los mapas de disparidad que se muestran. En las figuras 8(b) y 10 vemos como para *tsukuba* hay una aproximación a la solución correcta al aumentar el costo de oclusión. Para *corridor*, la aproximación también mejora con el aumento del costo de oclusión, sin embargo, en la primera fila (con potencia de ruido baja) la mejora no es tan apreciable.

Para el algoritmo GC, el comportamiento frente a la potencia del ruido agregado tiene un comportamiento parecido, que se comprueba tanto en las gráficas (figuras 7(c) y 7(d)) como en los mapas de

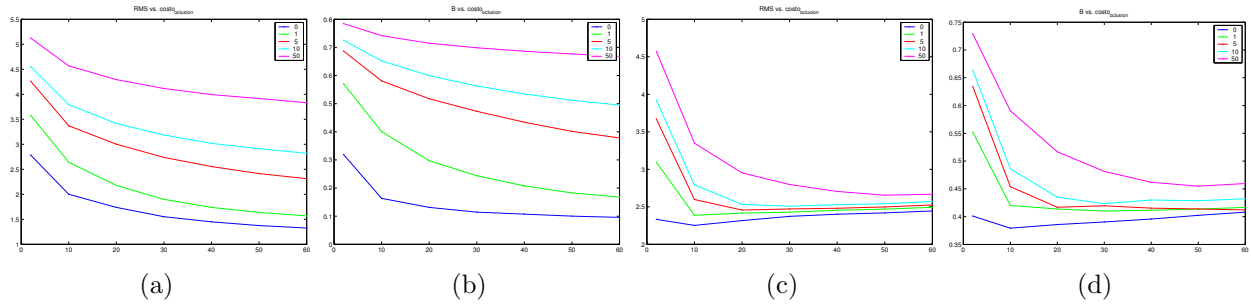


Figura 6: Resultados obtenidos para la con DP variando el costo de oclusión, paramétrico en la potencia del ruido agregado a las imágenes. (a) RMS en *tsukuba* (b) B en *tsukuba* (c) RMS en *corridor* (d) B en *corridor*

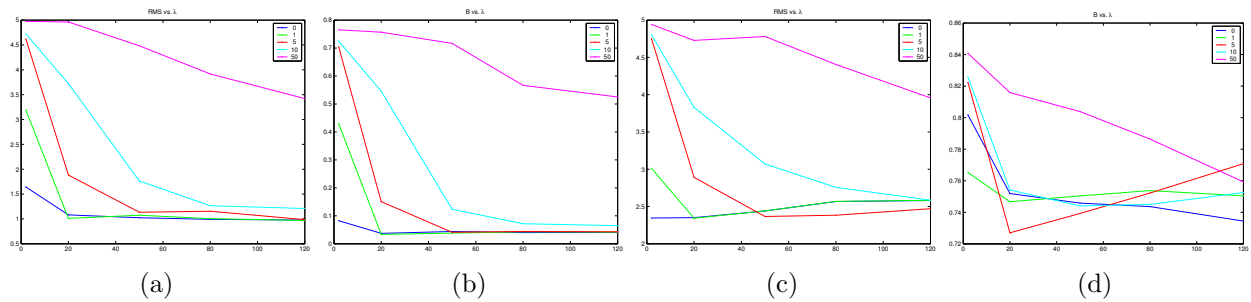


Figura 7: Resultados obtenidos para la con GC variando el costo de oclusión, paramétrico en la potencia del ruido agregado a las imágenes. (a) RMS en *tsukuba* (b) B en *tsukuba* (c) RMS en *corridor* (d) B en *corridor*

disparidad (figuras 8(a) y 9). Es relevante en este algoritmo la dependencia con el parámetro λ ; no se puede afirmar que la forma en que varían los resultados a las distintas pruebas tenga un comportamiento tan claro como en el caso de DP. Aquí los resultados son más irregulares (figuras 4, 7(c) y 7(d)). Los términos $E_{smooth}(\cdot)$ y $E_{oclu}(\cdot)$ se les asigna un valor proporcional a λ . Esto implica que al aumentar λ se tenderá a que la solución de mínima energía presente mayor suavidad (*smoothness*), pues se penaliza la poca suavidad de la solución; y será mas costoso etiquetar un punto como oculto. Esto hace que las regiones tiendan a hacerse más homogéneas (ver figura 4). Al aumentar λ el ruido deja de tener tanto peso en la minimización, y pesa más tener discontinuidades en el mapa de disparidad, por lo que la aproximación es mejor. De la mano de este comportamiento, el número de puntos que se etiquetan como ocultos disminuye, al haber menos planos que se solapan en las soluciones (ver figura 4). Encontrar una solución mediante el método de corte de grafos es similar a encontrar una superficie que se adapte a los datos de entrada de forma de minimizar la energía propuesta. Por esto la solución de este algoritmo se puede interpretar como un conjunto de parches de superficies planas (mismo valor de etiqueta–disparidad) que guardan una coherencia local. Los cambios entre los valores de estos parches dependen de la penalización en términos de la energía que representen.

Desde el punto de vista del número de puntos mal clasificados (B) en los experimentos realizados, *tsukuba* tienen un comportamiento como el que se reporta en la literatura referida. Con DP y sin ruido agregado el porcentaje de puntos con disparidad mal calculada varía de un máximo de 32% con un costo de oclusión bajo, llagando a un 10% con costos de oclusión medios y altos (ver figura 6(b)). Por otro lado, GC en el mismo experimento no supera el 5% de puntos con disparidad mal calculada, excepto con un valor de λ muy bajo que obtiene un 8%. Cuando se agrega ruido aumentando la potencia del mismo, este porcentaje con DP aumenta drásticamente, aún con costos de oclusión altos superando el 50%. GC tiene un comportamiento diferente que DP pues con valores de λ medios y altos con porcentajes de error en la disparidad menores al 10% (esto puede verse en la última columna de la figura 9). Con *corridor* la situación es diferente, siendo DP quien presenta un comportamiento

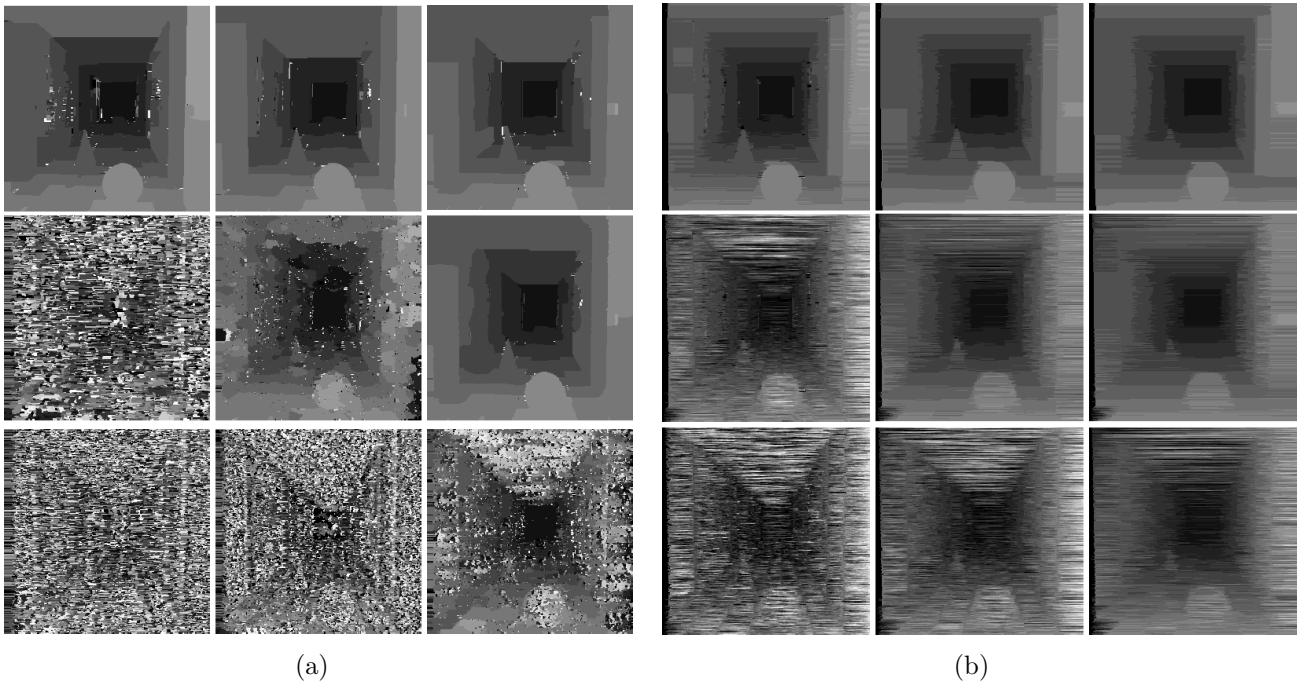


Figura 8: (a) Mapas de disparidad obtenidos para la escena *corridor* con GC. Cada columna corresponde a la salida con el mismo λ aumentando la potencia del ruido. Cada fila corresponde a la salida con la misma potencia de ruido aumentando λ . (b) Mapas de disparidad obtenidos para la escena *corridor* con DP. Cada columna corresponde a la salida con el mismo costo de oclusión aumentando la potencia de ruido. Cada fila corresponde a la salida con la misma potencia de ruido aumentando el costo de oclusión.

más regular ante la variación de los parámetros. El porcentaje de error en el caso de DP no supera el 50% en la mayoría de los experimentos realizados, mientras que GC no logra disminuir del 70% en ninguno de ellos. En las figuras 8(a) y 8(b) se observan los mapas de disparidad generados y se ve que los mayores problemas se dan que el plano inclinado no es bien capturado por los algoritmos. Sin embargo, DP tiene mejor comportamiento que GC en este caso.

En la figura 11 vemos un caso de interesante análisis en la zona del brazo de la lámpara. GC crea una región falsa uniendo ambos brazos y asignando una disparidad promedio a ésta región. Por su parte, DP mantiene ambos brazos separados y asigna una disparidad *razonable* entre los mismos. Este fenómeno, se debe a que GC intenta colocar una superficie plana en cada valor de disparidad diferente, penalizando la variación en las disparidades. El uso de un parámetro global λ en GC podría causar problemas en diferentes partes del mapa de disparidad. Tampoco DP logra un buen resultado en este caso, pero la solución es más próxima a la real.

Las disparidades obtenidas para *corridor*, ponen de manifiesto uno de los principales problemas de los algoritmos de cálculo de disparidad, la poca resolución que se obtiene en la disparidad. De la disparidad real, vemos que presenta un suave cambio a lo largo del corredor, lo cuál no es recuperado por ninguno de los métodos testeados. Tanto GC como DP obtienen cuantificaciones gruesas de la disparidad real, resultando en malos valores de *RMS* y *B*.

Para evaluar el grado de aproximación de la variación de la disparidad a lo largo del corredor, se tomó una región del techo del corredor que se muestra marcado en la figura 12(a). Este perfil de disparidades se grafica en la figura 12, junto con los perfiles de disparidades medios calculados por cada algoritmo, variando su parámetro. Se muestran los perfiles para las pruebas realizadas con ambos algoritmos, con y sin ruido agregado a las imágenes estéreo. Se verifica que ninguno de los algoritmos logra aproximar la pendiente correcta. Sin embargo DP se aproxima más a la pendiente

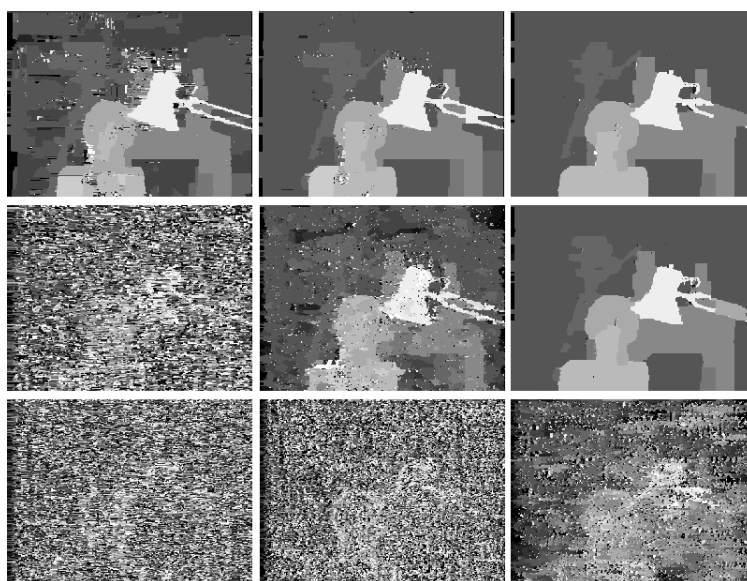


Figura 9: Mapas de disparidad obtenidos para la escena *tsukuba* con GC. Cada columna corresponde a la salida con el mismo λ aumentando la potencia de ruido. Cada fila corresponde a la salida con la misma potencia de ruido aumentando λ .

real que GC, cuando no se tiene ruido agregado. Cuando se le suma ruido, continúa teniendo una mejor aproximación pero presenta mayor dispersión.

5. Conclusiones

Ninguno de los algoritmos es óptimo para cualquier tipo de escenas y pueden presentar errores en algunas partes del mapa de disparidad. Como sucede con muchos algoritmos, el funcionamiento es óptimo con cierta estructura geométrica de la escena, pero cuando alguna de las restricciones en la estructura o hipótesis de la escena no se verifican, se generan errores. GC plantea una energía que se adapta correctamente a una estructura de superficies planas paralelas al plano de la imagen, pero que con imágenes reales, con planos con pendiente y ruido, puede generar problemas. DP parece ser un poco más robusto a estos problemas pero necesita una etapa de agregado de coherencia inter-scanline para poder dar resultados más aproximados. Las oclusiones presentes en las imágenes son detectadas, dependiendo del modelo utilizado, pero no hay una solución para la correcta detección de las mismas.

Desde el punto de vista de la complejidad, sin duda, GC presenta mayor complejidad que DP, tanto en la estructura de datos que debe manejar, y sobre la cuál debe realizar la minimización, como en el tiempo y número de operaciones para resolver el problema. El tiempo de ejecución de DP es varias decenas de veces menor que GC. En las pruebas que se realizaron, los tiempos promedio fueron de 450 mseg. para DP contra 29.2 seg. para GC, promedio en unas 140 pruebas con cada algoritmo.

En cuánto a la complejidad para seleccionar los parámetros para una buena solución, la implementación que dan los autores de GC incluye un modo donde λ se calcula automáticamente para las imágenes de entrada, obteniendo buenos resultados. Pero si se desea hacer una configuración «personalizada» la forma en que influye la variación de los parámetros en los resultados es más sencillo DP. Dependiendo del uso del mapa de disparidad, la resolución y precisión de la misma, podrán inclinar la decisión por uno u otro, si la escena se puede aproximar por planos paralelos a las cámaras y regiones grandes, la coherencia local que plantea GC le da mayores ventajas, además de dar un mapa de disparidad con poco ruido, y regiones bien definidas. En cambio si existen regiones con planos inclinados, o direcciones importantes angostas (como los brazos de la lámpara en *tsukuba*), DP puede ser considerado un buena opción.

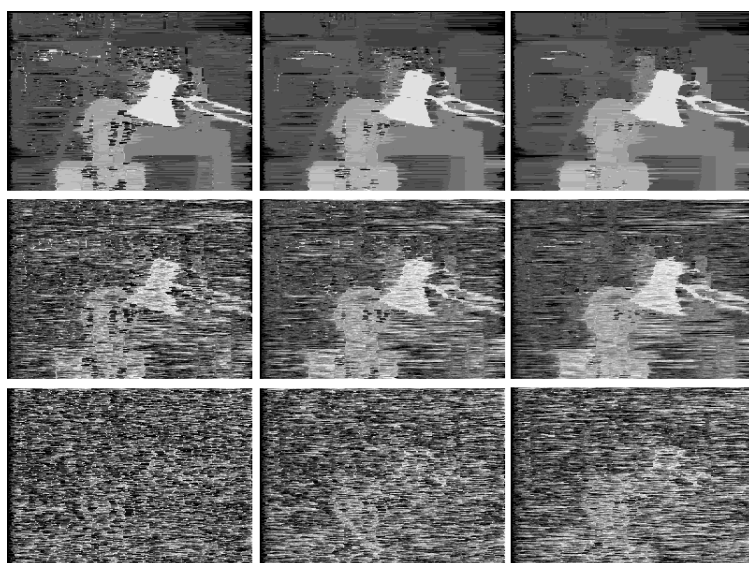


Figura 10: Mapas de disparidad obtenidos para la escena *tsukuba* con DP. Cada columna corresponde a la salida con el mismo costo de oclusión aumentando la potencia de ruido. Cada fila corresponde a la salida con la misma potencia de ruido aumentando el costo de oclusión.

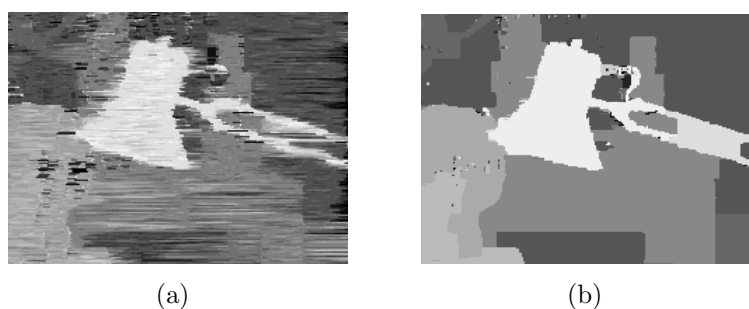


Figura 11: Detalle de los mapas de disparidad calculados. Mapas de disparidad calculados a partir del mismo par de imágenes ruidosas, con DP y GC con valores de costo de oclusión y λ medios. (a) DP. (b) GC.

A pesar que se pueden encontrar fallas y posibles mejoras, los resultados globales con imágenes sin agregado de ruido y con valores de los parámetros razonablemente configurados, dan al algoritmo GC como el que presenta los mejores resultados, no solo frente a DP, sino frente al resto de los algoritmos que mejores resultados han tenido en cálculo de disparidad [1]. Sin embargo, no presenta buenas soluciones para cualquier tipo escena y los tiempos de cálculo pueden llegar a ser muy restrictivos.

Referencias

- [1] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, pages 7–42, April-June 2002.
- [2] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc R Soc Lond*, pages 301–328, Mayo 1979.
- [3] Aaron F. Bobick and Stephen S. Intille. Large Occlusion Stereo. *International Journal of Computer Vision*, 33(3):181–200, September 1999.

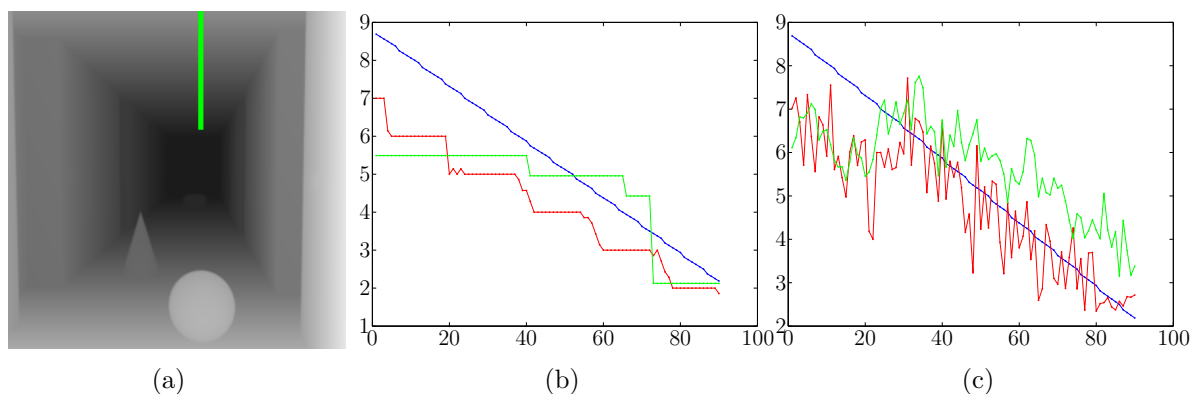


Figura 12: (a) En verde se muestra los puntos de la imagen que definen el perfil de disparidad evaluado. (b) En azul el perfil de disparidad extraído, en verde el perfil medio calculado por GC y en rojo, idem para DP. Sin ruido agregado. (c) Igual a (b) pero con ruido.

- [4] Pascal Fua. Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities. In *IJCAI*, pages 1292–1298, 1991.
- [5] Yuichi Ohta and Takeo Kanade. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, March 1985.
- [6] P.N. Belhumeur. A Bayesian Approach to Binocular Stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996. ISSN:0920-5691.
- [7] Vladimir Kolmogorov and Ramin Zabih. Computing Visual Correspondence with Occlusions via Graph Cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.
- [8] Sébastien Roy and Ingemar J. Cox. A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem. In *International Conference on Computer Vision*, pages 492–502, 1998.
- [9] Chris Buehler, Steven J. Gortler, Michael F. Cohen, and Leonard McMillan. Minimal Surfaces for Stereo. In *European Conference on Computer Vision (3)*, pages 885–899, 2002.
- [10] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, August 2003.
- [11] T.Kanade, K.Oda, A.Yoshida, M.Tanaka, and H.Kano. Video-Rate Z Keying: A New Method for Merging Images. Technical Report CMU-RI-TR-95-38, The Robotic Institute, Carnegie Mellon Institute, Diciembre 1995.
- [12] R. Shukla, H. Radha, and Martin Vetterli. Disparity dependent segmentation based stereo image coding. In *ICIP*, volume 1, pages 757–760, 2003.
- [13] Luigi di Stefano, Massimiliano Marchionni, and Stefano Mattoccia. A fast area-based stereo matching algorithm. *Image Vision Comput.*, 22(12):983–1005, 2004.
- [14] D. Scharstein and R. Szeliski. High-Accuracy Stereo Depth Maps Using Structured Light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, 1:195–202, June 2003.
- [15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.