Semi-Global-Matching modificado para visión estéreo en sistemas de tiempo real

S. Ibarra Delgado[a], J. Flores Troncoso[a], R. Sandoval Árechiga[a], I. A. Arriaga Trejo[b], J. Villanueva Maldonado[b], y J. Simón Rodríguez[b]

[a]*CIDTE, Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica.*
Av. López Velarde No. 801, Zacatecas, Zac, 98000, México.
http://www.uaz.edu.mx/
[b]*Cátedra CONACyT, CIDTE, Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica.*
Av. López Velarde No. 801, Zacatecas, Zac, 98000, México.
http://cidte.uaz.edu.mx/

2015 Published by $DIFU_{100}ci@$ http://difu100cia.uaz.edu.mx

Resumen

En la actualidad, es cada día más común el uso de sistemas de visión estereoscópica en sistemas de tiempo real, especialmente, aplicados a vehículos autónomos no tripulados. La necesidad de poder responder a la velocidad adecuada, para que este tipo de sistemas pueda reaccionar ante posibles obstáculos, implica que los algoritmos utilizados para encontrar la disparidad estéreo, vean afectada su precisión. El algoritmo Semi Global Matching (SGM), ha mostrado ser eficiente en términos de precisión, en el cálculo del mapa de disparidad. Sin embargo debido a la cantidad de recursos que necesita, su implementación en sistemas de tiempo real sigue siendo complicada. En este artículo presentamos una modificación al algoritmo SGM, en el que se encuentra la cantidad mínima de información ?global? necesaria para poder implementar este sistema eficientemente.

Palabras clave: Semi-Global-Matching, Tiempo Real, Visión estereoscópica.

1. Introducción

Los métodos de correspondencia estereoscópica tienen la capacidad de poder extraer información tridimensional de una escena. Esto lo hacen en basados en encontrar a que pixel en la imagen de referencia (izquierda generalmente), corresponde el pixel en la imagen objetivo (derecha). Para lograr esto, la mayoría de los algoritmos, toman información de su entorno para identificar cual región en la imagen de referencia se corresponde mejor con alguna región de la imagen objetivo.

Los algoritmos que tratan de encontrar esta mejor correspondencia son generalmente divididos en dos tipos [1]. Los algoritmos locales, los cuales encuentran la correspondencia tomando una región de soporte de tamaño relativamente pequeño en la imagen fuente y comparan la información que hay a su alrededor,

con una región de soporte de aspecto similar en la imagen objetivo, esto lo hacen desde una disparidad 0, hasta una disparidad maxima d_{max} , cuyo valor generalmente es el 10 % del tamaño máximo de la imagen. La disparidad seleccionada es aquella que tiene el valor mínimo. El otro tipo de algoritmo son los denominados algoritmos globales, estos generalmente enfocan la resolución del problema desde un punto de vista de optimización, de tal modo que en base a una función de energía, ver ecuación (1), encuentran la solución optima para una disparidad en un pixel dado. Para el problema de optimización, la asignación óptima de etiquetas, puede ser formulada como un problema de minimización de energía sobre un campo aleatorio de Markov. La función de energía esta compuesta típicamente de dos términos: un término unitario E_d que penaliza la inconsistencia con el dato observado, y un término compuesto de suavidad E_s que favorece la coherencia espacial entre las etiquetas.

$$\{I_p\} = \underset{I_p \in L \times L \times L}{\operatorname{argmin}} \left\{ \sum E_d(I_p) + \sum_{p,q} E_s(I_p, I_q) \right\} \quad (1)$$

El flujo de proceso que típicamente sigue un algoritmo estéreo se puede observar en la Figura 1. El proceso toma como origen un par de imágenes rectificadas, las cuales están alineadas horizontalmente, de este modo el problema se reduce a encontrar la correspondencia de un pixel en la imagen de referencia, en la misma linea de la imagen objetivo. Inicialmente las imágenes pasan por un pre proceso de filtrado para reducción de ruido y suavizado. En el primer estado se calculan las medidas de costo, aquí se utilizan medidas de similitud ya sean del tipo paramétricas o no paramétricas, como son el caso de las diferencias absolutas o la transformada del rango respectivamente. En el segundo estado las medidas de costo son agregadas sobre la region de soporte seleccionada, este estado tiene como objetivo agregar información sobre el vecindario del pixel que se evalúa, de tal modo que la medida de costo encontrada es mas confiable. El siguiente estado calcula la disparidad por medio de un método de optimización local o global. Finalmente, existe un último paso donde se refinan los valores de disparidad obtenidos, eliminado aquellas disparidades que no pudieron ser correctamente calculadas debido a los diferentes problemas que se tienen en las imágenes estéreo, como pueden ser entre otros, ruido debido a cambios radiométricos, zonas con patrones repetitivos o zonas con baja textura. Los métodos locales ponen mayor énfasis en los estados 1 y 2 del proceso, mientras los métodos globales se enfocan en el estado 3 del proceso.

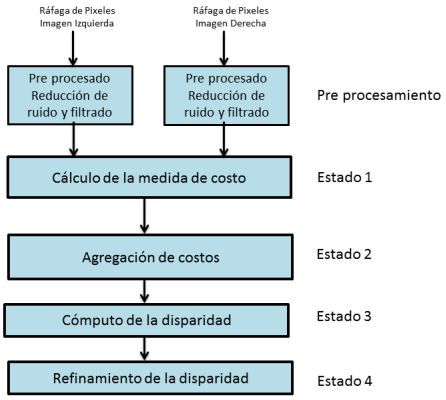

Figura 1. Estados básicos para el procesamiento de algoritmos estéreo.

Los algoritmos globales normalmente presentan mejores resultados que los algoritmos locales en términos de precisión. Desafortunadamente, estos algoritmos resultan costosos en términos de recursos de hardware que consumen y del tiempo que necesitan para su ejecución. Por otro lado los algoritmos locales muestran en términos de precisión, resultados más pobres, sin embargo, por la cantidad de información que manejan, son más susceptibles de poder responder de acuerdo a las restricciones impuestas por los sistemas de tiempo real.

Algunos autores [1, 2], han presentado alternativas que pretenden estar entre los dos extremos, con esto tratan de lograr un equilibrio entre los recursos y tiempo de ejecución contra la precisión de los mapas de disparidad. La propuesta de [2] es basada en programación dinámica y tiene como objetivo encontrar el camino que minimice los costos en la línea de referencia, desafortunadamente los resultado obtenidos por estos métodos generan un afecto de barrido al tener solo una línea como referencia el efecto puede ser apreciado en la Figura 2. Una alternativa a este problema es el presentado por [3] en este artículo, los autores realizan una optimización lineal pero en varias direcciones dando mucho mayor robustez al algoritmo y obteniendo resultados más halagadores en términos de precisión. La efectividad de esta propuesta, puede ser observada en el hecho de que los algoritmos basados en este método, se encuentran posicionados en los primeros lugares del ampliamente conocido set de

middlebury stereo [4].

Figura 2. Mapa de disparidad Tsukuba con programación dinámica.

2. Semi global matching

Cuando se realiza un cálculo de empate pixel a pixel, este puede resultar ambiguo, es común que se calculen falsos empates con valores de costo bajos y que estos sean los seleccionados como disparidades correctas, esto sucede generalmente por la presencia de ruido en la imagen o efectos radiométricos que pueden aparecer en las imágenes. Normalmente los métodos globales añaden una restricción extra relacionada con la suavidad, la cual tiene como objetivo penalizar los cambios bruscos en la imagen, lo que está altamente relacionado con los bordes de las mismas. Lamentablemente como se ha mencionado, el evaluar esta función en todo el espacio 2D de la imagen tiene el problema del alto consumo de tiempo de cómputo y la gran cantidad de recursos de hardware utilizados, además de que generalmente se causa un cuello de botella por el excesivo tráfico que hay con la memoria. Lo anterior hace muy difícil la implementación de los algoritmos globales en aplicaciones que tienen que ver con sistemas de tiempo real.

El método denominado Semi-Global-Matching [3], presenta como base la idea de empatar pixeles de imágenes estereoscópicas, haciendo múltiples aproximaciones 1D con lo que pretende lograr una aproximación global 2D. Los autores proponen una función de energía, ecuación (2), basada en términos de la medida de costo y la restricción de suavidad que penaliza los cambios de disparidad en las vecindades del pixel.

$$E(D) = \sum_p \Big\{ C(p, D_p) + \sum_{q \in N_D} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \Big\} \quad (2)$$

Donde el primer término, es la suma de todos los costos de empates para las disparidades de D. El segundo término suma una constante P_1 para todos los pixeles q en el vecindario N_p de p, para las cuales las disparidades tienen un ligero cambio (aproximadamente 1 pixel). El tercer término, agrega una constante P_2, que penaliza cambios de disparidad más grandes. El uso de una penalización pequeña, favorece superficies inclinadas o curvas. El uso de constantes para cambios grandes preserva las discontinuidades [5]. Las discontinuidades son perceptibles como cambios de intensidad. Esto se puede hacer adaptando P_2 a la intensidad del gradiente, esto es, $P_2 = \frac{P'_2}{|I_{bp} - I_{bq}|}$, para el vecindario de pixeles p y q en la imagen base I_b, sin embargo se debe de asegurar que siempre $P_2 > P_1$.

El problema de empate estéreo se puede formular como encontrar la disparidad D que minimice la función de energía $E(D)$. Sin embargo como el ámbito es 2D y es una minimización global, el problema es NP-completo para muchas discontinuidades. Como alternativa se presenta la idea de agregar múltiples costos de empate en espacio 1D. El costo agregado de suavidad $S(p, d)$ para un pixel p y disparidad d es calculado sumando los costos en las diferentes rutas que terminan en el pixel p en la disparidad d, esto se muestra en la Figura 3. Las rutas son proyectadas en líneas rectas en la imagen base, pero como líneas no rectas dentro de la correspondiente imagen de disparidad, de acuerdo a los cambios de disparidad en las rutas.

El costo $L'_r(p, d)$ a través de una ruta en la dirección r de un pixel p a una disparidad d es definido recursivamente tal como se muestra en la ecuación (3).

$$L'_r(p, d) = C(p, d) + \min \Big\{ L'_r(p - r, d), L'_r(p - r, d - 1) + P_1, L'_r(p - r, d + 1) + P_1, \min_i \{ L'_r(p - r, i) + P_2 \} \Big\} \quad (3)$$

La medida de costo utilizada propuesta originalmente por los autores del método es, Mutual Information. Sin embargo el método ha sido probado con diferente medidas de costo con buenos resultados en [6] se utiliza una

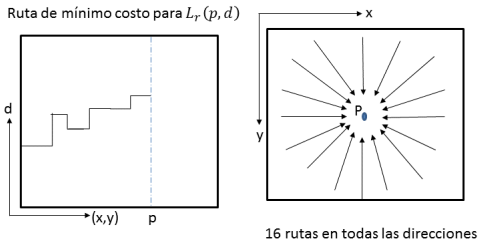

Figura 3. Direcciones de búsqueda de SGM y ruta mínima.

medida combinada con diferencias absolutas y transformada census, sus resultados en cuanto a precisión la colocan en los primeros lugares de [4]. A continuación se tiene un término que suma el costo más bajo del pixel previo $p - r$ de la ruta, incluyendo la penalidad para discontinuidades. El valor de L' se encuentra en permanente incremento a través de la ruta, con lo cual se pueden generar valores muy grandes, de tal modo que la expresión (3) puede ser modificada substrayendo el mínimo costo de la ruta del pixel previo de todo el término de la ecuación (4). La modificación no cambia la ruta actual a través del espacio de disparidad, dado que el término sustraído es constante para todas las disparidades para el pixel p. Esto es la posición del mínimo no cambia. Sin embargo, el límite superior puede ser dado por $L < C_{max} + P_2$.

$$\begin{aligned} L_r(p,d) = C(p,d) + \min\Big\{ & L_r(p-r,d), L_r(p-r,d-1) \\ & + P_1, L_r(p-r,d+1) \\ & + P_1, \min_i L_r(p-r,i) + P_2 \Big\} \\ & - \min_k L_r(p-r,k) \end{aligned} \tag{4}$$

Los costos L_r de todas las rutas son sumadas para todas las direcciones r, ver ecuación (5).

$$S(p,d) = \sum_r L_r(p,d) \tag{5}$$

El número de rutas que proponen los autores originales de SGM es de 6, sin embargo se han presentado propuestas [6] con menos rutas que presentan buenos resultados.

3. SGM modificado

En la presentación original del algoritmo SGM se proponen 16 rutas. Teniendo en cuenta que el algoritmo toma la información partiendo del volumen de disparidades de las imágenes, la cantidad de costos que deben de ser evaluados para encontrar el mínimo dada una posición dada, suponiendo una imagen cuadrada es $num_{costos} = 8 \times longitud_{linea}$. Si suponemos una imagen de 1024×1024 con 64 niveles de disparidad, el volumen de disparidades es de 64M y deben de ser evaluados 8k de medidas de costo por cada pixel de tal modo que para encontrar el mapa de disparidad para esta imagen es de 8G, suponiendo una imagen en escala de grises que ocupa solo un byte y que se desean procesar 30 cuadros por segundo, el total de medidas computadas por segundo, debe de ser de 240 GB/s. Esta cantidad difícilmente puede ser alcanzada, especialmente cuando se desea montar sobre dispositivos autónomos móviles. Algunos autores [7], han presentado variantes proponiendo menos rutas y mostrando resultados relativamente buenos, comparados con la propuesta original y obteniendo velocidades que permiten su implementación en tiempo real. Sin embargo estas implementaciones sufren el problema de que el mapa de disparidad es sesgado hacia las direcciones que son probadas.

En nuestra propuesta, el problema de manejar la gran cantidad medidas de costo que son necesarias para evaluar la disparidad en un pixel, es direccionando desde otro punto de vista. En lugar de eliminar rutas de búsqueda, lo que se busca es evaluar que tanta profundidad debe de tener la ruta a evaluar para obtener un resultado confiable. La idea proviene de, que dado que el método SGM es acumulativo, es decir, para decidir el costo, para una ruta dada, en el pixel $i + 1$ es necesario conocer su costo en i, y para que esta medida sea confiable, ¿cuántos costos anteriores deben de ser conocidos?

Nuestra propuesta tiene como primer objetivo encontrar cual es la profundidad necesaria en el cálculo del SGM dada una imagen, para esto se dividirá el estudio en dos zonas, zona de baja textura y zona texturizada. Las pruebas fueron realizadas con la imagen de prueba 190 del set 2012 dado por [8], que se muestra en la Figura 4.

De la imagen presentada en la Figura 4 se han seleccionado dos regiones una (1) en la que se muestra una zona de baja textura y otra (2) una zona claramente texturizada.

Se obtiene una máscara de la zona poco texturizada

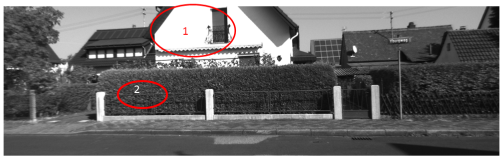

Figura 4. Imagen de prueba 190 KITTI visión Benchmark Suite 2012.

Figura 5 y se procede a realizar la evaluación solo sobre una ruta del SGM, en este caso $L0$. Los pixeles que se evalúan son los que se encuentran en la zona texturizada (negro) ya que la evaluación en la zona poco texturizada (blanca) falla por tener una gran cantidad de mínimos iguales.

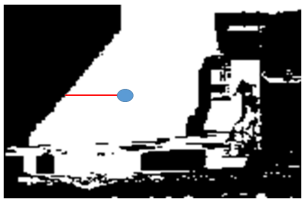

Figura 5. Zona de baja textura imagen 190.

En la Figura 6, se puede observar como al ir aumentando la profundidad de la zona de evaluación, se va obteniendo un mínimo cada vez más pronunciado hasta que es claramente visible. Cuando la profundidad evaluada es corta, de uno a cuatro pixeles, no es posible identificar un mínimo absoluto, pero conforme esta profundidad va creciendo, se puede observar cada vez en mínimo claramente distinguible. A partir de seis pixeles de profundidad y hay un mínimo que se puede distinguir, sin embargo este es más notorio con una profundidad de ocho.

Cuando el pixel a evaluar se encuentra en una zona de baja textura, como en la zona (2) de la imagen de referencia, se puede observar en la Figura 7 que con una profundidad muy pequeña se encuentra un mínimo absoluto claramente identificable.

Las pruebas anteriores sugieren que si se pueden identificar zonas texturizadas y no texturizadas claramente, entonces el algoritmo de SGM no necesita hacer una evaluación tan profunda en cada una de las rutas

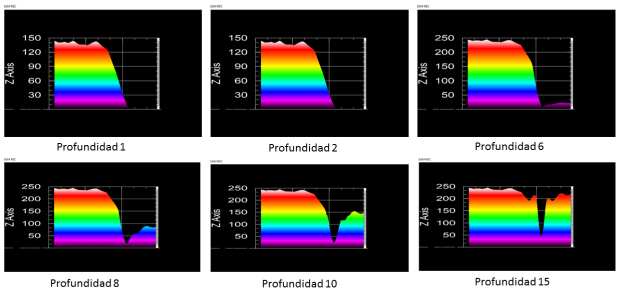

Figura 6. Cálculo de la disparidad en zona no texturizada, mínimos aumentando la profundidad.

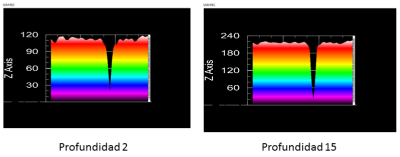

Figura 7. Cálculo de la disparidad en zona texturizada, mínimos aumentando la profundidad.

del mismo, pudiendo reducir las necesidades de cómputo notablemente, en el caso de estudio el número de costos a ser evaluados se reduce a $num_{costos} = 8 \times 15 = 120$, de tal modo que el número de costos a evaluar para asegurar un tiempo de respuesta que pueda ser considerado de tiempo real (30 cuadros por segundo) es de 3.54 GB/s.

4. Resultados

Para evaluar la efectividad de la modificación propuesta, se obtuvo el mapa de disparidad para la imagen de prueba. Se calculó el mapa de disparidad utilizando el método original con 8 rutas y nuestra propuesta también con 8 rutas. Utilizando el cálculo de zonas de baja textura propuesto en [9] se obtuvo la máscara de zonas de baja textura, la profundidad evaluada fue de 15 pixeles, los resultados pueden ser observados en la Figura 8.

Se puede observar que los dos mapas se comportan de manera similar en las zonas texturizadas, en las zonas poco texturizadas el algoritmo original se comporta mejor que el propuesto, al medirlos se obtiene solo un 5.34 % de degradación del original con respecto al propuesto.

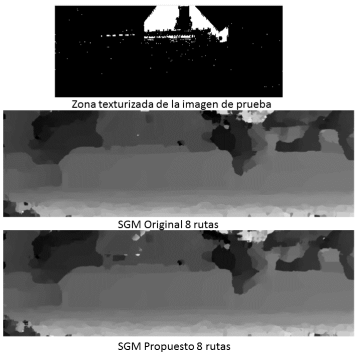

Figura 8. Evaluación SGM.

5. Conclusiones y trabajo futuro

De acuerdo a los resultados que se obtuvieron en esta investigación, se puede concluir que en el algoritmo de SGM, no es necesario recorrer toda una ruta para encontrar el costo acumulado mínimo, basta con recorrer una distancia corta de pixeles confiables para lograr un costo mínimo confiable. Lo anterior conlleva a una reducción de significativa del cómputo necesario para obtener una disparidad, de 220 GB/s a 3.54 GB/s, esta reducción permite que el algoritmo pueda ser implementado en sistemas de tiempo real.

Los resultados muestran que el algoritmo propuesto se comporta similar al algoritmo original en zonas texturizadas y solo ligeramente inferior en zonas poco texturizadas. Sin embargo esta degradación no es causada por las suposiciones hechas en el algoritmo, estas son debidas a que el algoritmo que identifica las zonas de baja textura obtiene falsos positivos. Entonces no es posible hacer una selección solo basados en la textura, existen otros factores que deben de ser tomados en cuenta como son las zonas ocluidas y los patrones repetitivos. Por lo anterior mencionado, en el futuro nosotros trabajaremos en buscar un clasificador binario que permita elegir correctamente las zonas confiables y las zonas no confiables. Con lo que el presente algoritmo pueda tendrá un mejor comportamiento

Referencias

[1] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” *IJCV*, vol. 47, no. 13, pp. 7-42, Apr. 2002.

[2] Jae Chul Kim; Kyoung Mu Lee; Byoung Tae Choi; Sang Uk Lee, “A dense stereo matching using two-pass dynamic programming with generalized ground control points,” *Computer Vision and Pattern Recognition*, 2005. CVPR 2005. IEEE Computer Society Conference on , vol.2, no., pp.1075,1082 vol. 2, 20-25 June 2005

[3] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE TPAMI*, 30(2):328?341, 2008.

[4] http://vision.middlebury.edu/stereo/, 2013.

[5] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts,” *IEEE Trans.* Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, 2001

[6] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang and X. Zhang, “On building an accurate stereo matching system on graphics hardware,” *Computer Vision Workshops* (ICCV Workshops), 2011 IEEE International Conference on, pp.467,474, 6-13 Nov. 2011.

[7] Banz, C.; Hesselbarth, S.; Flatt, H.; Blume, H.; Pirsch, P., “Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation,” in *Embedded Computer Systems* (SAMOS), 2010 International Conference on , vol., no., pp.93-101, 19-22 July 2010

[8] http://www.cvlibs.net/datasets/kitti/eval_stereo.php

[9] S. Ibarra Delgado, J. Flores Troncoco, H. Gamboa Rosales, R. Soule de Castro, “Algoritmo para la detección de zonas poco texturizadas en imagenes reales, para su uso en sistemas de visión estereoscópica,” *DIFU100ci@*, Universidad Autónoma de Zacatecas, ISSN 2007-3585, Vol. 7, No. 1, mayo-agosto 2013.